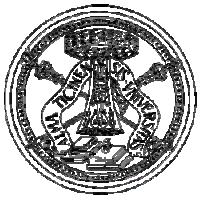


UNIVERSITA' DEGLI STUDI DI PAVIA
SEDE DISTACCATA DI MANTOVA
FACOLTA' DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

REALIZZAZIONE DI UN SERVER DI POSTA (POP3, IMAP E SMTP) CON PARTICOLARE ATTENZIONE AI FILTRI ANTISPAM

Laureando: Mariella Franciosi
Relatore: Prof. Giuseppe Federico Rossi

Anno Accademico 2004/2005

RINGRAZIAMENTI

Un particolare ringraziamento desidero porgerlo al Professor Giuseppe Federico Rossi
per la disponibilità e l'aiuto offerto per lo svolgimento del mio progetto.
Ringrazio inoltre Emanuele Goldoni e Luca Capisani
per l'appoggio e il prezioso supporto.
Infine un grazie anche alla mia famiglia, a Davide e a tutti i miei amici e
compagni di studi che mi sono stati vicini per raggiungere
questo importante traguardo.

INDICE

Pag.

1. INTRODUZIONE ALLA POSTA ELETTRONICA

La posta elettronica rappresenta ormai il mezzo di comunicazione asincrono più diffuso. I motivi di tale successo sono da ricondursi essenzialmente alla sua estrema facilità d'uso, velocità, accessibilità ed economicità. Permette di inviare messaggi a più persone contemporaneamente attraverso l'uso delle mailing list, di allegare documenti e consente lo scambio di informazioni multimediale: è possibile spedire, infatti, messaggi contenenti immagini, suoni, video, hyperlink, testo in formato HTML e applet Java.

Tuttavia anche la posta elettronica presenta dei limiti. Fra questi l'impossibilità di stabilire un contatto diretto con il destinatario, di avvisare dell'arrivo di un'e-mail, di garantire sicurezza nell'invio e soprattutto la difficoltà nell'identificazione del mittente. Questi ultimi due problemi sono stati in parte superati con l'introduzione della ricevuta di ritorno e con le applicazioni di firma elettronica. Il tallone d'Achille resta comunque la sicurezza.

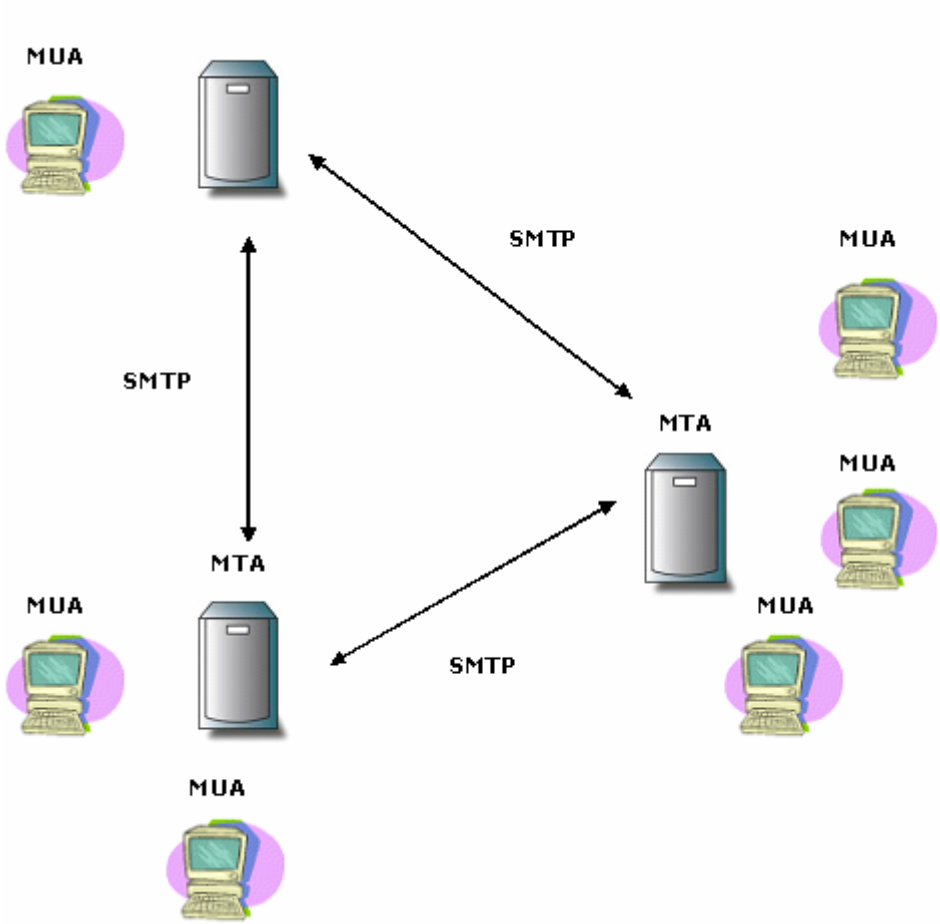
Lo scopo di questa tesi è quello di analizzare il funzionamento della posta elettronica e la sua principale debolezza che è rappresenta dalla minaccia che da diversi anni sta affliggendo gli utenti virtuali: lo spamming. Si focalizzerà quindi l'attenzione su questo fenomeno che sta diventando sempre più attuale per poi fornire un esempio pratico per provare a limitarlo.

Prima di tutto, spieghiamo in modo più approfondito come è composta e come funziona la posta elettronica.

1.1 COMPOSIZIONE

La posta elettronica si basa su un'architettura client-server con due ruoli: spedizione e ricezione. In particolare racchiude tre componenti principali, come si può osservare in figura:

- Il client → Mail User Agent (MUA);
- Il server → Mail Transfer Agent (MTA);
- Il protocollo SMTP (Simple Mail Transfer Protocol).



I MUA, anche chiamati mail reader, permettono agli utenti di leggere, inoltrare, salvare e comporre messaggi. In questi ultimi tempi si sono sviluppati soprattutto agenti dell'utente con GUI (Graphical User Interface), i quali attraverso interfacce grafiche permettono la composizione di messaggi multimediali. Tra questi i più diffusi sono Eudora, Notes, Microsoft Outlook, Netscape's Messenger.

I MTA, detti anche mail server, gestiscono la ricezione della posta dei loro utenti. Ogni utente ha una casella di posta (mailbox) che risiede nel server di posta. Inoltre l'MTA può effettuare lo smistamento verso altri server (relaying) cercando di evitare lo spamming. Oltre a contenere una mailbox per ogni utente il mail server si compone anche di una coda di messaggi (message queue) che contiene i messaggi da spedire agli altri server. I messaggi spediti e ricevuti sono memorizzati quindi sul server. Il server di posta deve essere sempre attivo, collegato ad internet e condiviso con altri utenti; è gestito dall'ISP (Internet Service Provider) che fornisce l'accesso alla posta all'utente.

SMTP è il protocollo su cui si basa la spedizione dei messaggi; trasferisce, infatti, i messaggi dal server di posta del mittente a quello del destinatario.

1.2 FUNZIONAMENTO

Le fasi di spedizione e ricezione delle e-mail seguono un determinato flusso di operazioni. In particolare lo user agent del mittente invia il messaggio di posta al suo server di posta dove viene accodato nei messaggi di posta in uscita. Dopodichè l'e-mail viene spedita al server di posta del destinatario che provvederà a collocarla nella rispettiva mailbox. Se il server del destinatario non è attivo allora il server del mittente collocherà il messaggio in una coda e riproverà a spedirlo solitamente ogni trenta minuti. Nel caso non dovesse riuscirci, dopo alcuni giorni il messaggio verrà cancellato e il server notificherà l'avvenuta rimozione al mittente. Per leggere i messaggi contenuti nella propria mailbox il destinatario dovrà accedere autenticandosi con username e password.
In qualche caso il mail server del mittente, detto SMTP server, coincide con il mail server del destinatario, detto POP server. Resta comunque il vincolo che i due servizi sono svolti su porte diverse.

1.3 FORMATO DEI MESSAGGI DI POSTA E MIME

Come i messaggi di posta tradizionali anche quelli di posta elettronica contengono delle informazioni quali indirizzo del mittente, del destinatario e data. Queste informazioni che nella posta cartacea sono esterne, qui vengono incluse nel messaggio in alcune linee di intestazione che precedono il corpo separate da un linea vuota (CRLF).
Il messaggio di posta elettronica si compone, infatti, essenzialmente di 2 sezioni, la sezione di intestazione e la sezione dedicata al corpo della mail. Altre sezioni di intestazione sono presenti in determinati casi, ad esempio se la mail contiene un allegato.

Le intestazioni contengono informazioni che vengono interpretate dal client di posta, per stabilire per esempio da chi è stata inviata la mail, a chi è diretta, se ci sono destinatari multipli e se sono presenti allegati. Inoltre alcune servono al client per stabilire il formato del corpo, il tipo di codifica di eventuali allegati e il formato stesso dell'allegato. Queste intestazioni non sono visibili quando leggiamo la nostra mail, ma alcuni dati ci vengono comunque proposti come il mittente e l'ora di invio.

Dopo il testo della mail troviamo la sezione dedicata agli eventuali allegati, i quali vengono inglobati nella mail e codificati con la tabella ASCII a 7 bit. Insieme ai bit dell'allegato troviamo altre informazioni come il nome del file e il suo tipo.

Il formato e il significato delle linee di intestazione vengono definiti dalla RFC 822. In particolare ogni linea è formata da una parola chiave seguita da due punti e da un valore (nome-header: valore-testuale). Fra queste alcune sono facoltative, altre obbligatorie come `From:` e `To:` che indicano

rispettivamente l'indirizzo del mittente e del destinatario. Fra le intestazioni facoltative la più importante è quella del `Subject:` che contiene l'oggetto del messaggio.

Vediamo un piccolo esempio:

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the meaning of life
Date: Wed, 15 Apr 1998 14:24:06 +0200
X-Mailer: Gorilla 3.2 (Win95; I)
```

`Date:` indica la data e l'ora di spedizione che può essere espressa in vari formati e `X-Mailer:` e in generale tutte le header inizianti per "X-" sono informazioni aggiunte dal client.

I messaggi di testo multimediali o dati non ASCII (caratteri usati da lingue diverse dall'inglese) contengono intestazioni extra non definite nella RFC 822. Si deve far riferimento alle RFC 2045 e 2046, nonché all'estensione MIME (Multipurpose Internet Mail Extensions) della RFC 822. Si tratta di un'estensione multi-uso del servizio di posta in Internet. Le intestazioni principali MIME sono sicuramente `Content-Type:` e `Content-Transfer-Encoding:`. La prima permette di specificare il tipo dell'oggetto multimediale inviato e il formato in modo che l'agente dell'utente possa compiere determinate azioni per interpretarlo. La sintassi dell'intestazione è la seguente:

```
Content-Type: type/subtype; parameters
```

I parametri, che sono modificatori del sottotipo, e il punto e virgola sono facoltativi, `type` indica il tipo generico di dato e `subtype` il formato specifico per quel tipo di dato. Ad esempio:

- `Content-Type: image/jpeg` significa che il messaggio contiene un'immagine con il formato jpeg e quindi fornisce all'agente dell'utente l'informazione che per visualizzare l'immagine dovrà decomprimerla secondo il formato del JPEG;.
- `Content-Type: text/html` significa che il messaggio contiene del testo in formato HTML e quindi avvisa l'agente dell'utente che per visualizzarlo dovrà interpretarne i tag.

Il MIME è stato progettato per essere ampliabile, infatti il set di coppie tipo/sottotipo con i parametri associati possono aumentare e per farlo in modo ordinato e pubblico secondo gli standard è stato istituito un processo di registrazione, descritto nella RFC 2048, che utilizza l'Internet Assigned Numbers Authority (IANA) come registro centrale.

Per quanto riguarda i tipi attualmente ne sono definiti sette e per ognuno di questi è presente un elenco di sottotipi. I principali tipi definiti sono: text, image e application.

Il tipo text (testo) specifica all'agente dell'utente destinatario che il corpo del messaggio contiene informazioni in formato testuale. Se viene abbinato con il sottotipo plain (`text/plain`) si tratta di testo normale senza particolari stili e formattazioni. Spesso vengono anche definiti i parametri

`charset=us-ascii` oppure `charset="ISO-8859-1"` che indicano il set di caratteri usato per generare il messaggio.

Il tipo image (immagine) specifica che il corpo di un messaggio è un'immagine e con il sottotipo indica il formato dell'immagine e quindi quale tecnica di decompressione dovrà utilizzare l'agente dell'utente destinatario per visualizzare l'immagine.

Il tipo application viene usato per dati che prima di essere utilizzati devono essere elaborati da una determinata applicazione. E' il caso per esempio di allegati come documenti di Microsoft Word e simili (`application/msword`).

L'intestazione `Content-Transfer-Encoding:` viene usata quando il messaggio contiene testo non ASCII e indica il tipo di codifica usato per trasformarlo in ASCII per il trasporto con SMTP. Questa informazione è molto utile per l'agente dell'utente destinatario che poi dovrà decodificare il messaggio usando l'equivalente decodifica.

Vediamo un piccolo esempio:

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding : base64
Content-Type: image/jpeg

(dati codificati base-64.....
.............................................
.....dati codificati base-64)
```

Notiamo che l'agente dell'utente del mittente usa la codifica base-64 per inviare un'immagine JPEG. Altra tecnica di codifica molto diffusa è la quoted-printable usata soprattutto per codificare messaggi in formato ASCII a 8 bit.

L'intestazione `MIME-Version:` indica la versione MIME che è stata usata.

Quando un messaggio di posta contiene più oggetti viene usato il tipo multiparte del MIME, in particolare sarà presente la seguente linea di intestazione:

`Content-Type: multipart/mixed`

Dato che la posta elettronica include tutti gli oggetti in un unico messaggio, sorge la necessità di determinare dove un oggetto inizia e dove finisce, la codifica adottata per ogni oggetto non ASCII e il tipo di contenuto di ciascun oggetto. Per ovviare a questo problema si introducono alcuni caratteri di confine (boundary character) per delimitare l'inizio e la fine di ciascun oggetto ed inoltre si specifica per ognuno di essi il `Content-Type:` e il `Content-Transfer-Encoding:`.

Riporto ora un esempio:

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe with commentary
MIME-Version: 1.0
Content-Type: multipart/mixed; Boundary=StartOfNextPart

--StartOfNextPart
Dear Bob,
Please find a picture of an absolutely scrumptious crepe.
 --StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
dati codificati base64……..
……
……..dati codificati base64
--StartOfNextPart
Let me know if you would like the recipe.
```

La linea di separazione inizia sempre con due trattini.

Anche il server SMTP del destinatario può inserire linee di intestazione; in particolare appende all'inizio del messaggio la linea `Received:`. Entrando più nel dettaglio viene indicato con `from` il nome del server SMTP che ha spedito il messaggio, con `by` il nome del server che ha ricevuto il messaggio e infine viene specificata l'ora in cui è stato ricevuto il messaggio. Ecco un esempio:

```
Received: from crepes.fr by hamburger.edu; 12 Oct 98 15:27:39 GMT
```

Solitamente un messaggio contiene più di una intestazione `Received:`. Questo indica che prima di giungere a destinazione il messaggio è passato per altri server oppure è stato inoltrato ad altri server. Ogni server infatti lascia una sorta di firma per indicare il fatto che il messaggio è transitato di lì e anche l'ora. Questo risulta molto utile per ricostruire la storia e il percorso di un messaggio. Inoltre il server del mittente attribuisce un `Message-ID` al messaggio generalmente nella forma <una-qualsiasi-stringa@nome-host>.

Altro campo è il `Return path:`, il quale contiene l'indirizzo per le risposte, viene utilizzato quando il mittente specifica un indirizzo diverso da quello di invio per ricevere le risposte; se non è specificato l'indirizzo per le risposte sarà utilizzato quello di invio. `Bcc`: viene usato per contenere gli indirizzi di altri eventuali destinatari che essendo nella lista Bcc (Blind Carbon Copy) riceveranno una copia della mail senza vedere tutti i destinatari ma vedranno solo il mittente.

Vediamo un esempio del contenuto completo (header più corpo) del sorgente di un messaggio posta elettronica con un allegato.

Indirizzo a cui inviare le mail di risposta, impostato dal client email del mittente:

```
Return-Path: < mittente@hotmail.com >
```

Elenco dei server SMTP che hanno ricevuto il messaggio e quando lo hanno ricevuto:

```
Received: from hotmail.com (65.54.245.56) by mail-1.tiscali.it (6.7.019) id
        3F335F020042BB6C for destinatario@tiscali.it; Wed, 20 Aug 2003 10:28:29
        +0200
Received: from mail pickup service by hotmail.com with Microsoft SMTPSVC;
        Wed, 20 Aug 2003 01:28:26 -0700
Received: from 80.105.15.38 by by1fd.bay1.hotmail.msn.com with HTTP; Wed,
        20 Aug 2003 08:28:25 GMT
```

Qui possiamo notare come il server "by1fd.bay1.hotmail.msn.com" (80.105.15.38) abbia inizialmente ricevuto il messaggio, successivamente lo ha instradato verso "hotmail.com" (65.54.245.56). Il server "mail-1.tiscali.it", MX per il dominio tiscali.it, riceve questo messaggio da "hotmail.com" per l'indirizzo destinatario@tiscali.it.
A questo punto il messaggio è giunto a destinazione ed è pronto per essere letto.

Sezione contenente i dati di invio e di ricezione del messaggio

```
X-Originating-IP: [80.105.15.38]
X-Originating-Email: [mittente@hotmail.com]
From: "mittente"
To: destinatario@tiscali.it
Bcc: Questa sezione contiene gli indirizzi inseriti nella BCC list
Subject: prova Oggetto della mail
Date: Wed, 20 Aug 2003 10:28:25 +0200 Ora di invio del messaggio
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary="----=_NextPart_000_18d8_2ee6_83a"
Message-ID:
```

Header custom del client di posta elettronica (Evolution) che ha scaricato la mail

```
X-OriginalArrivalTime: 20 Aug 2003 08:28:26.0367 (UTC)
        FILETIME=[06CDB0F0:01C366F5]
X-Evolution-Source: pop://destinatario@pop.tiscali.it/

This is a multi-part message in MIME format.
```

------=_NextPart_000_18d8_2ee6_83a
Content-Type: text/plain; format=flowed
Descrizione del testo della mail HTML o testo, in questo caso testo

Testo della mail (in questo caso in formato di testo semplice)

Testo della mail

Il testo che segue è stato inserito in automatico da hotmail

_____
STOP MORE SPAM with the new MSN 8 and get 2 months FREE*
http://join.msn.com/?page=features/junkmail

sezione dedicata agli allegati

------=_NextPart_000_18d8_2ee6_83a
Content-Type: image/jpeg; name="01-g.jpg" *nome del file*
Content-Transfer-Encoding: base64 *tipo di codifica*
Content-Disposition: attachment; filename="01-g.jpg"

Questa sezione è l'allegato codificato come sopra descritto

/9j/4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAAPAAA/+4ADkFkb2JlAGTAAAAAAf/b
AIQABgQEBAUEBgUFBgkGBQYJCwgGBggLDAoKCwoKDBAMDAwMDAwQDA4PEA8ODBMTFBQTExwbGxsc
Hx8fHx8fHx8fHwEHBwcNDA0YEBAYGhURFRofHx8fHx8fHx8fHx8fHx8fHx8fHx8fHx8fHx8f
Hx8fHx8fHx8fHx8fHx8f/8AAEQgB4AKAAwERAAIRAQMR....

Per motivi di spazio non è stato visualizzato l'intero allegato

....GORHGgCNCKVXnwoEgcVaThj/hQuTG57gP1OXEVSSRoO
K4LQkgyB/V21CgLh5hpUpnnQSKqOIIKgA45EchVCG1ZIndkEqBsBaRl7ez2UIQuH5eWP7qoZA0HL
E8UoUhchXiuVCNgLiuWC4UEkBcQHHE8efdQAPxJjzAoMCqoIIyPHjQgW5AZdgxyoUhfmviTQh//Z

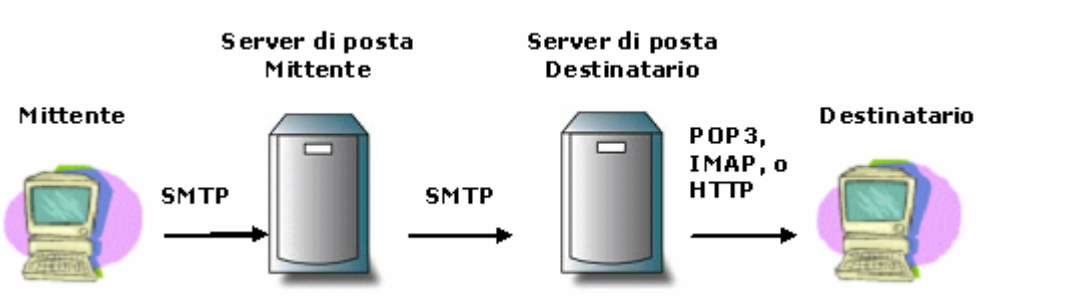
------=_NextPart_000_18d8_2ee6_83a--

Qui finisce l'allegato e anche la mail

1.4 PROTOCOLLI

Come già detto in precedenza, la posta elettronica per la spedizione dei messaggi si affida al protocollo SMTP. Anche l'agente dell'utente lato mittente, il quale non può comunicare direttamente con il MTA del destinatario, utilizza SMTP per trasferire i suoi messaggi al suo MTA. L'agente dell'utente lato destinatario invece non si può affidare a SMTP per recuperare i messaggi di posta, in quanto si tratta di un protocollo di push (letteralmente: spingi).

Per l'accesso e per la ricezione della posta sono stati quindi introdotti due protocolli di tipo pull (letteralmente: estrai): POP e IMAP. Per la posta elettronica basata sul web si adotta il più famoso protocollo HTTP.



1.4.1 Il protocollo SMTP

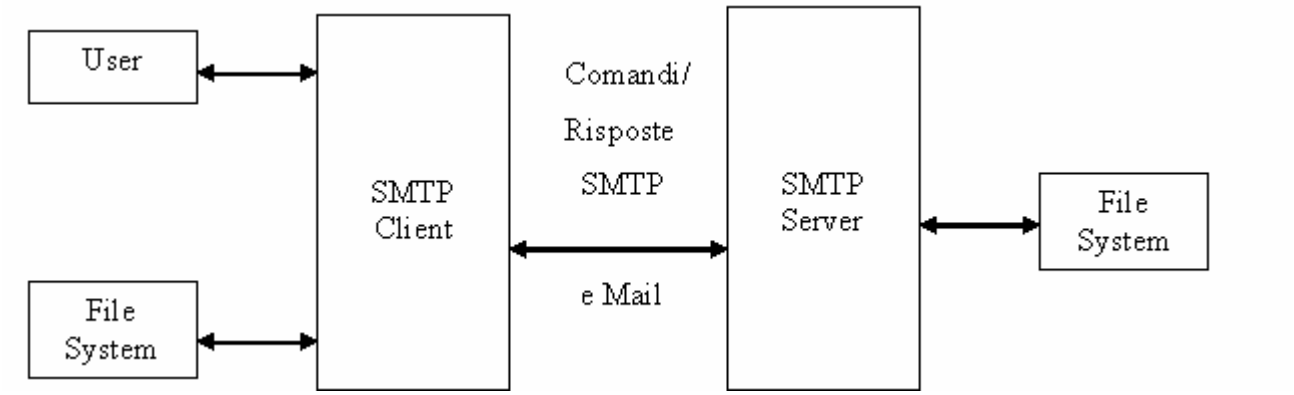
L'SMTP rappresenta il cuore ed è il protocollo principale dello strato di applicazione della posta elettronica. E' definito dalla RFC (Request for Comments) 821 che è stata poi sostituita dalla RFC 2821 datata agosto 1982. Vengono presi in considerazione anche altri standard come la RFC 822 che descrive la sintassi degli headers della mail e il formato del messaggio, la RFC 1049 che definisce le strutture dati per interpretare correttamente il contenuto delle mail e la RFC 974 che si occupa del routing delle mail tramite DNS. Lavora in stretta collaborazione con l'estensione MIME (Multipurpose Internet Mail Extensions) che definisce intestazioni extra per il formato dei messaggi, descritto nelle RFC che vanno dalla 2045 alla 2049.

Nonostante la sua vasta diffusione, purtroppo la tecnologia su cui si basa SMTP risulta a volte essere datata anche perchè definita nel lontano 1982. Supporta, infatti, solo il formato ASCII a sette bit per il corpo e le intestazioni dei messaggi. Questo poteva non essere un problema negli anni '80 ma adesso con il progresso raggiunto nella capacità e velocità di trasmissione e la possibilità di inviare messaggi multimediali con file audio, video o immagini questa restrizione diventa un limite. Per poter spedire un messaggio quindi è richiesto che i dati binari multimediali vengano tradotti in formato ASCII per poi essere decodificati in binario sul lato destinatario.

Il compito principale di SMTP riguarda il trasferimento delle e-mail tra i server di posta. Per fare questo si appoggia al servizio trasferimento affidabile dei dati del protocollo di trasporto TCP. Essendo un protocollo che sfrutta l'architettura client-server i server di posta vengono classificati di

volta in volta come client SMTP, se inviano messaggi di posta agli altri server, o come server SMTP se li ricevono.

Può essere schematizzato così:

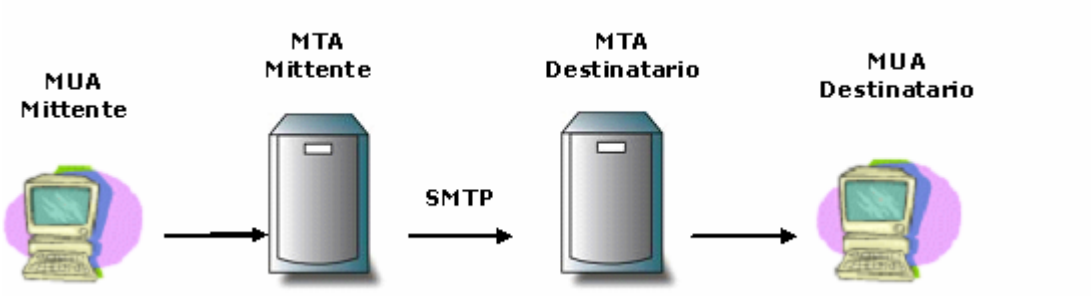


SMTP utilizza quindi il servizio affidabile TCP sulla porta 25 e solitamente effettua una trasmissione diretta fra server di posta mittente e destinatario non appoggiandosi a nessun server di posta intermedio.

La spedizione consiste in tre fasi:

1. handshaking (stretta di mano): apertura della connessione TCP, fase di presentazione necessaria per sincronizzare i server e prepararsi per il trasferimento del messaggio. In questa fase il client presenta l'indirizzo e-mail del mittente e del destinatario.
2. trasferimento del messaggio suddiviso a sua volta in:
   - inizio invio
   - elenco dei destinatari
   - invio del messaggio
3. chiusura: chiusura della connessione TCP. Trattandosi di una connessione persistente la chiusura della connessione avviene solo quando non ci sono più messaggi da spedire. Per ogni messaggio vengono ripetute le fasi di handshacking e trasferimento.

Vediamo ora più in dettaglio le operazioni base del protocollo SMTP, illustrate anche in figura:



a) Il mittente compone il messaggio attraverso il suo user agent e inserisce l'indirizzo del destinatario;
b) Lo user agent spedisce la mail al suo server di posta (MTA), il quale la colloca nella coda dei messaggi da inviare;
c) Il lato client SMTP, ovvero il server di posta del mittente, apre una connessione TCP con il lato server (server di posta del destinatario) sulla porta 25;
d) Dopo la fase iniziale di handshake, il client SMTP manda il messaggio al server;
e) Dopo aver ricevuto il messaggio, il server SMTP lo sistema nella casella di posta del destinatario;
f) Infine il destinatario invoca il suo user agent per leggere il messaggio.

Il client comunica con il server attraverso alcuni comandi in formato ASCII text e il server risponde con lo status code e una frase esplicativa. Per indicare la fine del messaggio viene inviata una linea con un punto, ovvero CRLF.CRLF dove CR sta per Carriage Return e LF per Line Feed rispettivamente.

Comandi del protocollo SMTP

I principali comandi usati dal protocollo SMTP sono:

HELO: Identifica il client SMTP al server SMTP, ovvero il computer che si sta rivolgendo al server.

MAIL: Richiede al server ricevente l'accettazione di un messaggio di email per l'inoltro.

MAIL FROM: <*indirizzo mittente*>: Indica la casella di posta del mittente, ovvero l'indirizzo di ritorno al quale riportare eventuali errori.

RCPT TO: <*indirizzo destinatario*> : Indica la casella di posta del destinatario (RCPT sta per *Recipient*). Si può indicare più di un destinatario ripetendo il comando RCPT TO.

DATA: Indica al server che quanto digitato successivamente saranno i dati del messaggio di posta, si segnala la fine del messaggio inviando una riga contenente solo un punto.

RSET: Annulla i comandi (*Reset*) precedentemente inviati nella sessione SMTP corrente.

VRFY <*stringa*>: Chiede al server se la stringa di testo immessa rappresenta un nome utente presente ed in tal caso visualizza l'intero indirizzo.

HELP: Visualizza i comandi disponibili sul server.

NOOP: Non esegue nessuna operazione restituisce solo un messaggio 250 (Ok) se il server risponde.

QUIT: Termina la sessione SMTP corrente.

Una sessione SMTP attraversa quindi almeno sei fasi:

1. Il client SMTP contatta il server sulla porta TCP 25. Se questo è in ascolto e la connessione è accettata risponde con un messaggio 220 (Ready);

2. Il client chiede di stabilire la sessione SMTP inviando il comando HELO seguito dal FQDN (Fully Qualified Domain Name). Se il server accetta risponde con un messaggio 250 (Ok);

3. Il client indica il proprio indirizzo tramite il comando MAIL FROM: <*indirizzo mittente*>. Il server risponde con 250 (Ok) per ogni destinatario accettato;

4. Successivamente il client indica al server i destinatari del messaggio tramite RCPT TO: <*indirizzo destinatario*> ed il server risponde per ogni destinatario accettato un codice 250 (Ok);

5. Il client comunica al server l'intenzione di scrivere il corpo del messaggio con DATA**.** Il server risponde con un codice 354 e indica come marcare il termine del messaggio. I campi come Date, Subject, To, Cc, From vanno inseriti tra i dati della mail;

6. Completato il messaggio da scrivere tramite **.** il server memorizza la mail. A questo punto è possibile, scrivere un nuovo messaggio oppure inviare il comando QUIT, dopo il quale il server invia i messaggi e risponde con un codice 221 (Closing) e la connessione TCP viene terminata.

Riporto ora un esempio di una sessione SMTP da linea di comando:

```
homer@Joker:~$ telnet smtp.springmail.it 25
Trying 195.130.225.171...
Connected to smtp.springmail.it.
Escape character is '^]'.
220 mail.springmail.it ESMTP Service (6.5.032) ready
Tramite il programma telnet viene contattato il server SMTP sulla porta TCP 25
HELO
250 mail.springmail.it Missing required domain name in HELO, defaulted to your
IP address [62.10.125.229]
Il client si identifica tramite il comando helo (non inviando un nome di
dominio, viene utilizzato l'indirizzo IP)
HELP
214-Valid SMTP commands:
214- HELO, EHLO, NOOP, RSET, QUIT, STARTTLS
214- MAIL, RCPT, DATA, VRFY, EXPN, HELP, ETRN
214-For more info, use HELP <valid SMTP command>
214 end of help
HELP QUIT
214-Syntax: QUIT
214-Purpose: request closing of the connection
```

```
214 end of help
Il client richiede una lista dei comandi disponibili tramite help e
successivamente un aiuto per il comando QUIT
MAIL FROM:<homer@springmail.it>
250 MAIL FROM:<homer@springmail.it> OK
RCPT TO:<arnaldo.z@email.it>
250 RCPT TO:<arnaldo.z@email.it> OK
DATA
Inizia l'inserimento del corpo della mail
354 Start mail input; end with <CRLF>.<CRLF>
From:Homer
To:Arnaldo
Subject:Il protocollo SMTP

Ciao,
questa mail serve per illustrare
il protocollo SMTP

--
Arnaldo aka [Homer]
.
250 Mail accepted
Il comando "." punto termina l'inserimento del messaggio; Il server SMTP lo
accetta
QUIT
221 mail.springmail.it QUIT
Connection closed by foreign host.
La connessione TCP termina e si ritorna al prompt
homer@Joker:~$
```

Il client di posta elettronica (KMail, Outlook, Mozilla Thunderbird) si occupa solitamente di comunicare con il server al posto nostro in base alla propria configurazione.

Confronto con HTTP

Particolarmente utile risulta confrontare il protocollo SMTP con il protocollo HTTP. Entrambi riguardano il trasferimento di file da un host ad un altro. In particolare HTTP trasferisce oggetti da un Web server ad un web client che ne ha fatto richiesta, mentre SMTP trasferisce e-mail da un server di posta all'altro. Inoltre usano tutti e due la connessione persistente.

Nonostante tutto presentano alcune differenze rilevanti. Per prima cosa HTTP è un protocollo pull (estrai) e la connessione TCP viene aperta dalla macchina che richiede gli oggetti. SMTP invece è

un protocollo di tipo push (spingi), la connessione TCP è aperta dalla macchina che vuole spedire il messaggio. Seconda differenza sta nella codifica ASCII: SMTP la richiede, HTTP non ha questa restrizione. Infine, HTTP usa un messaggio per ogni oggetto mentre SMTP incapsula tutti gli oggetti in un singolo messaggio.

1.4.2 Il protocollo POP

Il protocollo POP (Post Office Protocol, protocollo dell'ufficio postale) rappresenta il protocollo di accesso alla posta più diffuso. Ciò è dovuto alla sua estrema semplicità che però allo stesso tempo ne limita le funzionalità. E' definito dalla RFC 1939 e la versione a cui è giunto è la 3 e quindi verrà più facilmente trovato come POP3 (Post Office Protocol – versione 3). Per eseguire il processo di scaricamento delle mail viene instaurata una connessione TCP sulla porta 110. Come per SMTP una sessione POP consiste in uno scambio di comandi e informazioni tra il client (MUA del mittente) e il server (MTA del mittente). La sessione passa attraverso tre fasi:

1. Autorizzazione: dopo l'avvenuta connessione TCP il server presenta il suo messaggio di saluto e richiede al client il nome dell'utente e la password per autorizzare l'utente a scaricare la posta;
2. Transazione: durante questa fase vengono scaricati i messaggi e l'utente può inviare comandi per gestire i messaggi di posta. Più in dettaglio si possono contrassegnare messaggi per l'eliminazione, rimuovere il contrassegno, ottenere informazioni sulle statistiche della posta, creare cartelle dove spostare i messaggi e fare la ricerca di messaggi;
3. Aggiornamento: dopo aver inviato il comando `quit` per concludere la sessione, si passa a questa fase di aggiornamento; vengono eseguiti i comandi di cancellazione precedentemente memorizzati e successivamente la connessione TCP termina.

Durante le varie fasi il server mantiene le informazioni di stato come i messaggi contrassegnati per l'eliminazione, mentre durante le sessioni POP3 non le conserva e questo rende estremamente semplice l'implementazione di un server POP3.

E' possibile che su un server POP3 sia impostato un tempo di inattività, trascorso il quale si viene automaticamente disconnessi senza passare nella fase di aggiornamento. Questo significa che la connessione TCP è terminata e gli eventuali comandi di cancellazione impartiti al server non saranno presi in considerazione.

Ci sono due alternative che si prospettano dopo che i messaggi sono stati prelevati:

- dopo la copia i messaggi vengono rimossi dal server (download and delete, scarica e cancella);
- dopo la copia i messaggi restano sul server (download and keep, scarica e conserva).

Per la modalità download and delete sono previsti i comandi LIST, RETR e DELE.

Una problematica da non sottovalutare riguarda il fatto che usando questa tecnica se l'utente usa macchine diverse per scaricare la posta, i messaggi verranno sparpagliati su di esse mentre nella usando la tecnica download and keep i messaggi rimangono sul server e quindi l'utente può leggere i messaggi su macchine diverse senza che questi vengono cancellati dopo ogni accesso.

Comandi del protocollo POP

I principali comandi utilizzati dal protocollo POP3 sono i seguenti:

USER <*nomeutente*>: Identifica l'utente che si connette al server;

PASS <*password*>: Invia *in chiaro* la password dell'utente che si sta autenticando;

STAT: Restituisce il numero di messaggi presenti e lo spazio da essi occupato;

LIST <*numero messaggio*>: Senza parametri indica la dimensione di ogni messaggio, altrimenti solo quella del messaggio indicato;

RETR <*numero messaggio*>: Visualizza il messaggio indicato;

TOP <*numero messaggio*>: Visualizza un numero predefinito di linee dalla testa del messaggio;

DELE <*numero messaggio*>: Cancella dal server il messaggio indicato;

NOOP: Non esegue nessuna operazione restituisce solo un messaggio +OK se il server risponde;

RSET: Cancella le operazioni di cancellazione DELE in precedenza inviate al server;

QUIT: Termina la sessione POP3 corrente e si disconnette dal server.

Una tipica sessione POP3 si articola quindi in questo modo:

3. Il client POP3 si connette al server POP3 costantemente in ascolto sulla porta 110.
4. Una volta connesso il server invia un messaggio di saluto al client solitamente indicando il nome e/o la versione del software server.
5. Si passa quindi alla fase di autenticazione dove il client invia al server POP3 i comandi USER <*nomeutente*> e PASS <*password*>. Una volta che il client è stato autenticato si possono eseguire le operazioni per gestire la posta come leggere, cancellare un messaggio ecc.
6. E' possibile inviare i comandi al pop server il quale risponde con +OK in caso di comando eseguito correttamente, e con -ERR nel caso in cui non riesca ad interpretare il comando.
7. La sessione viene terminata con il comando QUIT.

Ecco un esempio di sessione POP3 da linea di comando:

```
homer@Joker:~$ telnet pop.springmail.com 110
Collegamento al pop server di nome pop.springmail.com tramite il programma
telnet sulla porta 110
Trying 213.92.5.75...
Connected to pop.springmail.com.
```

Escape character is '^]'.
+OK ifm-pop (version 4.0.0) at all-1.inet.it starting.
Il server risponde, ed è pronto per l'autenticazione, ora siamo in AUTHORIZATION state
USER homer@springmail.com
+OK Password required for homer@springmail.com.
Viene inviato il nome utente in questo caso rappresentato dall'intero indirizzo homer@springmail.com Il server risponde +OK, ovvero che ha accettato l'utente e richiede la password
PASS hmj2001np
+OK homer@springmail.com has 1 visible message (0 hidden) in 1173 octets.
Viene inviata la password ed il processo di autenticazione termina con successo. Ci troviamo ora in TRANSACTION state
STATS
-ERR Unknown command: "stats".
Inviando un comando errato, STATS anziché STAT, la risposta è un –ERR
STAT
+OK 1 1256
Stat viene eseguito con successo (+OK) ed indica che c'è un messaggio di 1256 byte
LIST
+OK 1 visible messages (1256 octets)
1 1256
.
LIST visualizza tutti i messaggi, è presente un solo messaggio
RETR 1
+OK 1256 octets
Return-Path:
Received: from [::ffff:193.70.193.55] by hal-5.inet.it via I-SMTP-4.3.7-430
id ::ffff:193.70.193.55+8UyItiMrJDJ; Wed, 02 Apr 2003 22:47:27 +0200
Received: from email.it (62.10.125.8) by mail1c.webmessenger.it (6.7.016)
(authenticated as arnaldo.z@email.it)
id 3E8868D4001242B0 for homer@springmail.com; Wed, 2 Apr 2003 22:47:24 +0200
Message-ID:
Date: Wed, 02 Apr 2003 22:47:21 +0200
From: Arnaldo Zitti
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.3)
Gecko/20030312
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: homer@springmail.com
Subject: Il protocollo POP3

```
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 8bit
X-UIDL: A0Y7UQ00N4rWVMr9ot4i
Salve,
questo è un messaggio di prova, come sample per
l'INFOBOX sul POP3 su OpenSkills.
--
Arnaldo aka [Homer]
.
Il comando RETR visualizza l'intero messaggio
DELE 1
+OK Message 1 has been deleted.
.
Viene cancellato il messaggio 1 con il comando DELE ed il server ne da conferma
In realtà il messaggio non viene cancellato ora, ma solo in UPDATE state
QUIT
+OK Pop server at all-1.inet.it signing off.
Una volta dato il comando QUIT si entra in UPDATE state, ed i messaggi verrano
realmente cancellati
Connection closed by foreign host.
homer@Joker:~$
La connessione TCP è terminata e si ritorna al prompt
```

Come per SMTP, durante una sessione POP3 tramite un client di posta elettronica come KMail, Outlook Express, Eudora od altri, è il client stesso che in base ai parametri di configurazione dell'account si occupa per noi di scambiare i messaggi con il server.

1.4.3 Il protocollo IMAP

Il protocollo IMAP (Internet Mail Access Protocol, protocollo di accesso alla posta Internet) è un protocollo di accesso alla posta che permette di compiere le elaborazioni direttamente sul server remoto centralizzato. In questo modo si può avere a disposizione tutti i messaggi indipendentemente dalla macchina con la quale ci si connette. Al contrario POP3 scarica i messaggi in locale e non permette di creare cartelle remote e anche per questo motivo è stato introdotto il protocollo IMAP come sostituzione del POP3. Presenta infatti una maggiore flessibilità ed un sistema di gestione più esteso e complesso.

E' definito dalla RFC 2060 e la versione a cui è giunto è la 4 e quindi verrà più facilmente trovato come IMAP4 (Internet Mail Access Protocol – versione 4). E' più complesso di POP3 ma nello stesso tempo ha più funzionalità. Il server IMAP all'arrivo di ogni messaggio lo associa alla cartella

INBOX del destinatario, che poi lo gestirà secondo le sue esigenze. Come POP3 anche IMAP permette agli utenti di creare cartelle e spostare i messaggi ma offre l'ulteriore funzionalità di crearle sul server remoto e quindi anche la ricerca viene effettuata su server remoto. A differenza di POP3, IMAP mantiene le informazioni tra uno sessione e l'altra. Altra caratteristica da non sottovalutare è la possibilità di scaricare solo alcune componenti del messaggio; questo si rivela molto utile soprattutto quando la connessione è a bassa larghezza di banda.
Riassumendo le funzionalità offerte da IMAP sono:
- Accesso e gestione dei messaggi di posta direttamente sul server;
- Possibilità di creare, cancellare e rinominare mailbox;
- Supporto della modalità di lavoro offline per i client e successiva sincronizzazione quando possibile;
- Ricerca e selezione dei messaggi in base ad attributi quali testo o contenuto MIME.

Una sessione IMAP consiste in una serie di comandi inviati dal client al server, il quale resta in ascolto sulla porta TCP 143, e terminati da CRLF (*Invio*). Ogni comando ha un identificatore, cioè un prefisso alfanumerico (Es. A0001) chiamato tag che lo precede, il cui compito è quello di far corrispondere le risposte del server alle rispettive domande del client.
Una sessione IMAP consiste quindi nelle seguenti fasi:

1. Instaurazione della connessione TCP da parte del client con il server IMAP (in ascolto sulla porta 143);
2. Risposta del server con il messaggio di benvenuto;
3. Autenticazione;
4. Gestione dei messaggi di posta da parte dell'utente attraverso il client;
5. Fine della connessione al seguito della richiesta da parte client o per scadenza del tempo di timeout;

Comandi del protocollo IMAP

Avendo la possibilità di gestire la posta in modo centralizzato sul server, e potendo accedere quindi da diversi macchine, IMAP implementa un sistema di attributi in grado di indicare lo stato corrente di un messaggio. Per ogni messaggio vengono definiti uno o più flag, che iniziano con il carattere "\" e ne indicano lo stato:
\Seen: indica che il messaggio e' stato letto;
\Answered: indica che e' stata inviato un messaggio di risposta;
\Flagged: indica che sul messaggio e' stato impostato un flag urgent o Special Attention;
\Deleted: indica che il messaggio e' stato contrassegnato per la cancellazione;

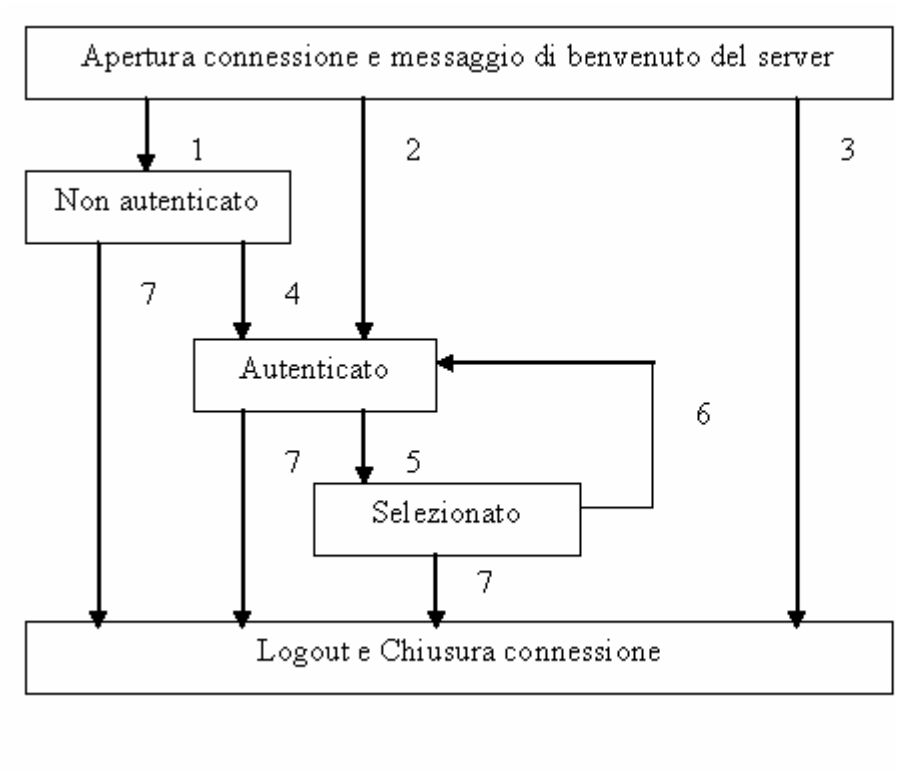
\Draft: indica che il messaggio non e' completo ed e' quindi contrassegnato come bozza;

\Recent: indica che il messaggio e' appena giunto nella casella di posta.

Una sessione IMAP può attraversare quattro stati:

1. Non autenticato;
2. Autenticato;
3. Selezionato;
4. Disconnesso (Logout);

Possono essere così rappresentati:



1) Connessione senza preventiva autenticazione

2) Connessione con preventiva autenticazione

3) Connessione respinta

4) Login riuscita

5) Comando Select eseguito con successo oppure Comando Examine

6) Comando Close command o Select fallito oppure Comando Examine

7) Logout e Chiusura Connessione o interruzione del server

In relazione allo stato della sessione, sono disponibili determinati comandi per l'elaborazione dei messaggi.

Comandi nello stato Non autenticato

Nello stato Non autenticato, si trovano gli utenti che sono collegati al server ma che si devono ancora autenticare. I comandi disponibili in questo stato sono quindi quelli necessari per l'autenticazione:

AUTHENTICATE: indica il metodo di autenticazione da utilizzare;

LOGIN <*utente*> <*password*>: invia al server il nome utente e la password per autenticarsi;

Comandi nello stato Autenticato

Una volta forniti i corretti valori di login e password si passa allo stato Autenticato. E' possibile scegliere, esaminare, creare, cancellare, rinominare una mailbox, controllarne lo stato e gestire le sottoscrizioni ad altre caselle. I comandi utilizzati sono:

SELECT <*mailbox*>**:** seleziona la casella di posta con cui lavorare;

EXAMINE <*mailbox*>: seleziona la casella di posta ma con accesso in sola lettura;

CREATE <*mailbox*>: crea una nuova casella di posta;

DELETE <*mailbox*>: cancella una casella di posta;

RENAME <*vecchio-nome-mailbox*> <*nuovo-nome-mailbox*>: rinomina una casella di posta;

SUBSCRIBE <*mailbox*>: aggiunge la casella di posta alla lista delle caselle attive da visualizzare in un client;

UNSUBSCRIBE <*mailbox*>: elimina la casella di posta alla lista delle caselle attive da visualizzare in un client;

LIST <*riferimento*> <*mailbox*>: restituisce un sottoinsieme dei nomi disponibili al client;

LSUB <*riferimento*> <*mailbox*>: restituisce un sottoinsieme delle caselle sottoscritte dall'utente;

STATUS <*mailbox*> <*stato*>: indica lo stato della casella di posta;

APPEND <*mailbox*> <*messaggio*>**:** aggiunge un messaggio alla casella di posta selezionata.

Comandi nello stato selezionato

Si passa allo stato selezionato quando si e' scelto su quale casella lavorare. Sono disponibili quindi i comandi per la gestione dei messaggi:

CHECK: funzione di manutenzione del server IMAP;

CLOSE: riporta nello stato Autenticato e rimuove i messaggi con flag \Delete attivo;

EXPUNGE: rimane nello stato Selezionato e rimuove i messaggi con flag \Delete attivo;

SEARCH *[setcaratteri] <parametri>:* cerca i messaggi che soddisfano i parametri di ricerca indicati;

FETCH <*messaggi*> <*dati*>*:* visualizza determinati dati di un messaggio (Es. Subject, data, testo ecc);

STORE <*messaggi*> <*valore-dati*>: cambia i dati che si voglio modificare nel messaggio. Viene solitamente usato per cambiare i flag di stato;
COPY <*messaggi*> <*mailbox*>*:* copia i dati nella casella di posta specificata;
UID <*comando*> <*argomenti*>: esegue i comandi impartiti in base al numero Uid (Unique Identifier) anziché il numero del messaggio.
Comandi nello Stato Disconnesso (Logout)
Una volta che il client effettua una richiesta di chiusura della connessione, si passa nello stato Disconnesso, per terminare il collegamento.
Comandi disponibili in ogni stato
Alcuni comandi non sono associati ad un particolare stato del server e quindi sono sempre disponibili:
CAPABILITY*:* indica le funzioni supportate dal server;
NOOP: non esegue nulla, risponde in modo positivo se il server è attivo. Utilizzato per evitare la disconnessione causa timeout;
LOGOUT: indica al server che il client vuole terminare la connessione.

Ecco un esempio di sessione IMAP tramite telnet:

```
[homer@Enigma homer]# telnet imap.joker.net 143
* OK [CAPABILITY IMAP4REV1 LOGIN-REFERRALS STARTTLS AUTH=LOGIN] joker.net
IMAP4rev1 2001.315 at Sun, 13 Jul 2003 22:09:17 +0200 (CEST)
Il server risponde e visualizza la versione del protocollo. Siamo in Non-
Authenticated State.
a100 LOGIN homer onslls
a100 OK [CAPABILITY IMAP4REV1 IDLE NAMESPACE MAILBOX-REFERRALS SCAN SORT
THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND] User homer authenticated
Viene eseguito il login e l'utente è accettato. Siamo in Authenticated State.
a101 select inbox
Viene selezionata la mailbox inbox (la posta in arrivo). Siamo in Selected
State.
* 2 EXISTS
* NO Trying to get mailbox lock from process 925
* 0 RECENT
* OK [UIDVALIDITY 1058126897] UID validity status
* OK [UIDNEXT 3] Predicted next UID
* FLAGS (NonJunk \Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (NonJunk \* \Answered \Flagged \Deleted \Draft \Seen)]
Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/homer
```

```
a101 OK [READ-WRITE] SELECT completed
La prima voce 2 EXISTS indica che sono presenti due messaggi
a102 fetch 1:2 (flags body[header.fields (subject)])
Viene effettuata una richiesta di visualizzazione dei subject delle due mail
presenti...
* 1 FETCH (FLAGS (\Seen NonJunk) BODY[HEADER.FIELDS ("SUBJECT")] {37}
Subject: Protocollo IMAP Overview
)
* 2 FETCH (FLAGS (\Seen NonJunk) BODY[HEADER.FIELDS ("SUBJECT")] {20}
Subject: Prova 2
)
a102 OK FETCH completed
a103 FETCH 1 (body[text])
Richiesta del corpo del messaggio 1
* 1 FETCH (BODY[TEXT] {105}
Messaggio di prova,
relativo all'Infobox sul protocollo IMAP per Openskills
--
Arnaldo aka Homer
)
a104 STORE 1 +FLAGS (\Deleted)
Il messaggio 1 viene marcato con il flag \Deleted ovvero da cancellare
* 1 FETCH (FLAGS (\Seen \Deleted NonJunk))
a104 OK STORE completed
a105 EXPUNGE
Vengono cancellati i messaggi con il flag \Deleted attivo (in questo caso il
messaggio 1)
* 1 EXPUNGE
* 1 EXISTS
* 0 RECENT
a105 OK Expunged 1 messages
a106 LOGOUT
Una volta lanciato il comando logout si entra in Logout State
* BYE joker.net IMAP4rev1 server terminating connection
a106 OK LOGOUT completed
Il server invia il messaggio BYE e termina la connessione
```

Come per gli altri protocolli a livello applicativo, un client di posta che supporta IMAP si occuperà per l'utente di scambiare i messaggi con il server.

POP vs. IMAP

Riassumiamo in una tabella che confronta il protocollo POP con il protocollo IMAP le caratteristiche di entrambi i protocolli.

	POP	**IMAP**
Scaricamento della posta	Utile se l'utente scarica la posta con una sola macchina	Utile se l'utente scarica la posta con diverse macchine
Carico sul server	Normale	Elevato
Gestione scaricamento	Non permette la gestione della mailbox sul server	Permette la gestione della mailbox sul server
Complessità	Minima	Medio-alta
Diffusione	Elevata	Più recente, poco diffuso
Sincronizzazione	Richiede una sincronizzazione completa	Si può ricorrere a servizi centralizzati
Flessibilità	Ristretta	Elevata
Sicurezza	Per leggere i messaggi bisogna necessariamente scaricare tutto il messaggio con gli evidenti rischi	Si possono visualizzare solo alcune componenti del messaggio (per esempio solo le intestazioni), maggior sicurezza
Uso on-line/off-line	Più comodo, si può gestire la posta off-line	Per gestire la posta è necessario essere on-line
Connessione	Più comodo per la connessione dial up	Più comodo per la connessione continua

1.4.4 Posta elettronica basata sul web - HTTP

Recentemente si sta diffondendo la possibilità e la comodità di spedire messaggi e accedere alla posta elettronica attraverso il proprio browser web. In questo caso l'agente dell'utente risulta essere un browser web e l'accesso alla posta avviene con il protocollo HTTP. In altre parole, quando un utente scarica la posta, i messaggi vengono trasferiti dal server di posta al browser dell'utente

usando HTTP al posto di POP o IMAP. Per quanto riguarda l'invio di un messaggio, il browser spedisce la mail al suo server di posta attraverso HTTP invece di utilizzare il protocollo SMTP. I server di posta continuano comunque a comunicare con SMTP. Le cartelle possono essere gestite sul server remoto come avviene anche per IMAP. In molti casi infatti le implementazioni della posta elettronica basata sul web utilizzano un server IMAP per permettere ai loro utenti di sfruttare le funzionalità che offre IMAP per la gestione delle cartelle.

2. INTRODUZIONE ALLO SPAM

Letteralmente la parola spam si riferisce ad un particolare tipo di carne di maiale in scatola della Hormel Foods Corporation. Il termine inglese deriva infatti da *spiced ham.*

Col passare del tempo ha assunto un altro significato più comune che spesso viene attribuito ad una scenetta comica dei Monty Python, un gruppo comico inglese degli anni ’70 molto conosciuto. In un episodio della serie televisiva "Monty Python's Flying Circus" un uomo e la moglie entrano in bar e chiedono il menù alla cameriera. L’elenco delle pietanze presenta sempre spam: uova e spam, uova pancetta e spam, uova pancetta salsiccia e spam, spam uova spam spam pancetta e spam e così via. La moglie però non desidera affatto lo spam. Nel frattempo da un tavolo alle spalle della coppia un gruppo di vichinghi con elmi cornuti ogni volta che la cameriera propone un piatto contenente lo 'spam', iniziano a cantare "spam, spam, spam..." in modo sempre più forte insistente e fastidioso, fino al culmine del finale dove il canto prende il sopravvento su tutto.

Il termine spam quindi grazie a questa scenetta è stato associato ai messaggi di posta non richiesti che per la loro invasività e per la loro frequenza vengono spesso mescolati con gli altri messaggi di posta e limitano la possibilità di comunicare. In particolare questi messaggi indesiderati presentano la caratteristica di essere ripetitivi e inviati in grandi quantità. La gravità dello spam non deriva dal contenuto ma dal fatto di essere inviato anche se non richiesto alla propria casella di posta che è proprietà privata.

Diamo una definizione più precisa: *Spam su Internet è uno o più messaggi non richiesti, inviati o postati come parte di un più grande insieme di messaggi, tutti aventi contenuto sostanzialmente identico.*

Si possono quindi percepire da questa definizione due caratteristiche fondamentali per classificare la posta come spam: la molteplicità e la non consensualità. Se un messaggio indesiderato è inviato ad una sola persona quindi non si ricade nella definizione. Inoltre l’utilizzo legittimo della posta elettronica è consentito solo se consensuale. In altre parole, il destinatario deve autorizzare l’invio esplicitamente o implicitamente. L’abuso si verifica quindi quando un indirizzo viene usato per scopi al di fuori di quello per cui è stato reso pubblico l’indirizzo. A volte stabilire se un messaggio è consensuale non è del tutto scontato. A questo proposito può essere utile porsi la domanda “Perché questo messaggio è stato mandato proprio a me?” per capire se il messaggio in questione risulta essere non richiesto.

Lo spam è un vero e proprio furto di servizi. Infatti si commette furto di servizi quando si utilizzano per i propri scopi computer e risorse altrui che sono in rete e vengono mantenuti per fare altre cose. In altre parole, si tratta di usare risorse altrui contro la volontà, espressa o tacita, di chi ha il diritto

di escluderlo. Ed è per questo motivo che la condanna allo spam è diventata ormai a livello mondiale.

2.1 TIPOLOGIE DI POSTA INDESIDERATA E DI SPAMMER

E' possibile collocare le email indesiderate in cinque diverse categorie:

- Hoax, ovvero le bufale e le catene di Sant'Antonio
- Worm, email mandate da virus
- UCE, Unsoliticed Commercial Email
- UBE, Unsoliticed Bulk Email
- Messaggi derivanti da iscrizioni a mailing list

Per quanto riguarda le prime due categorie è necessario sensibilizzare l'utenza al problema e quindi diffidare delle email a contenuto pietoso che richiedono di inoltrare il messaggio creando una crescita esponenziale nella divulgazione del testo ed installare e aggiornare regolarmente software antivirus.

La mailing-list consente a chiunque di iscriversi e spesso non richiede nemmeno la conferma diretta all'utente e quindi è facile trovarsi iscritti senza saperlo. Per fortuna questo fenomeno che aiuta il diffondersi di posta indesiderata sta scomparendo.

La fonte maggiore di spam è comunque l'UCE. Si tratta di email contenenti materiale commerciale che, senza richiesta ed autorizzazione vengono recapitate direttamente nelle mailbox degli utenti. Si spazia dal cibo, al materiale pornografico, servizi, software, ecc. Anche le UBE sono da considerare spam, in quanto riguardano un requisito essenziale per classificare le email come spam: la molteplicità. Si tratta infatti di email non richieste e mandate in grandi quantità.

Anche gli autori di questo inquinamento di posta spazzatura, ovvero gli spammer si possono classificare. Troviamo infatti:

- Junk Mailers o Junkers, coloro che inviano annunci commerciali non richiesti;
- Mailbombers, chi invia enormi quantità di email per inondare le mailbox di altri;
- Forgers, coloro che inviano messaggi inserendo indirizzi di posta di altri all'interno;
- Nuisances, chi vuole disturbare per qualche ragione o spesso anche senza.

2.2 SPAM IN PERCENTUALE

Secondo un rapporto datato giugno 2004 e diffuso da Brightmail, società attiva nella realizzazione di strumenti di sicurezza informatica lo spamming sta raggiungendo livelli ormai preoccupanti e le

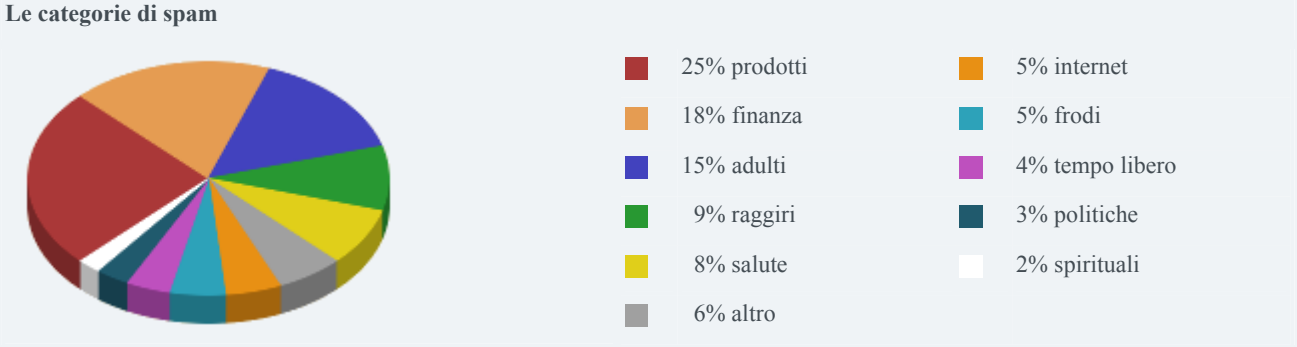
previsioni non rassicurano per niente. Si rischia di questo passo l'inutilizzabilità dello strumento simbolo della comunità Internet, ovvero la posta elettronica.

Tre anni fa il dato era fermo ad un'e-mail di spam ogni cento email legittime. Se solo si confrontano i dati di giugno 2003 con quelli di un anno dopo balza all'occhio la crescita smisurata del fenomeno. Prima i dati parlavano di un 49% di email indesiderate rispetto a tutta la posta ricevuta. Ora i dati sono più allarmanti: 65%, quasi i due terzi dell'intera posta mondiale in circolazione. A giugno 2004 la previsione per il 2005 era dell'80% e di poco si è discostata.

Secondo un recente rapporto semestrale di F-Secure Corporation, nei primi sei mesi del 2005 i risultati sono preoccupanti e rischiano di mettere in crisi l'intero sistema di posta mondiale. La percentuale di email indesiderate ha raggiunto infatti il livello record dell'85% dell'intero traffico mondiale. Importante notare che nonostante il predominio dello spam in lingua inglese nelle altre lingue ha già raggiunto il 26%. E' necessario correre ai ripari anche perché si teme che il fenomeno inizi a dilagare anche sulle tecnologie "mobile".

Allo stato attuale non è ancora disponibile una soluzione veramente efficace al problema. Non si parla solo di uno sforzo legislativo ma soprattutto tecnologico. Su questo fronte si stanno attivando i maggiori provider mondiali per cercare di contrastare il fenomeno a livello globale. Si prevede quindi un affinamento delle tecniche utilizzanti i filtri sia lato client sia lato server, nonché lo sviluppo di una soluzione comune, anche se riscontra notevoli difficoltà adattarla a tutti i provider del mondo. Per questo motivo l'Asta (Anti spam technical alliance), alleanza che include tutti i maggiori provider, sta cercando di risolvere il problema alla base, ovvero propone tecniche per identificare il mittente, in modo da poter bloccare gli spammer. In questa ottica è stato organizzato un incontro internazionale sullo spam, proposto dalla International Telecomunications Union (ITU), che fa capo all'ONU per cercare di arrivare ad una legislazione comune e studiare nuovi sistemi antispam. Le speranze di arginare il fenomeno se si lavora insieme sono ottimistiche: prevedono di piegare la piaga dello spam entro 24 mesi. Staremo a vedere…

2.3 CATEGORIE DI SPAM



Brightmail, ha diffuso anche dati riguardanti la composizione per argomento delle email indesiderate. Balza all'occhio la prevalenza negativa dell'e-commerce. Si piazzano al primo posto con ben il 25% infatti le offerte e le pubblicità di beni o servizi in vendita. Poco lontano troviamo, con il 18% le offerte o informazioni finanziarie, con il15% quelle a contenuto esclusivamente per adulti. Bufale e catene di Sant'Antonio con il 9% precedono salute, medicina e farmacia ferme all'8%. Il resto degli argomenti non supera il 5%: internet e informatica si fermano al 5%, così come truffe e frodi, tempo libero e hobby al 4%, propaganda politica al 3% e infine il 2% riguarda temi religiosi e spirituali. Il 6% si riferisce alle email immondizia che non hanno un tema specifico.
Confrontando le categorie rispetto a quelle di sei mesi prima si può notare la conferma del dominio dell'e-commerce, che guadagna un 3% a scapito di finanza e adulti, e la crescita delle bufale, dei raggiri e delle truffe. Ciò indica una tendenza sempre più massiccia ad utilizzare la posta per la pubblicità e per scopi illegali.

2.4 COSTI

Oltre a rappresentare una fastidiosa intrusione, da non sottovalutare sono anche le ingenti spese e i danni che l'inquinamento della posta sta provocando, che stanno raggiungendo ormai cifre da capogiro: si parla di 25 miliardi di dollari all'anno. Questo dato, fornito dall'ITU si basa su una stima dell'incremento, dovuto allo spam, del tempo di elaborazione, dello spazio di immagazzinamento, dei costi di utilizzazione della linea telefonica, dell'occupazione di banda, del tempo perso per fare pulizia nella propria mailbox, ecc.
Già a gennaio 2001 uno studio ordinato dalla Commissione europea e chiamato "Unsolicited Commercial Communications and Data Protection" valutava il costo per singolo utente in 30 euro all'anno. Più dettagliatamente citando quella fonte: "Su scala mondiale, assumendo una comunità online di 400 milioni di utenti internet, il costo globale dello scaricamento di messaggi pubblicitari usando l'attuale tecnologia può essere cautelativamente stimato in dieci miliardi di euro all'anno - e questa è solamente la frazione di costo che viene sostenuta dagli utenti direttamente".
Anche i provider devono sostenere ingenti costi a causa dell'intasamento provocato dalle email indesiderate. Devono aumentare la banda, la potenza e la memoria per i loro sistemi, nonché più personale per il controllo e il customer-care e per individuare misure tecnologiche per contrastare virus, tentate truffe, messaggi e immagini inadatte ai minori. Purtroppo questi costi dovranno trasferirsi anche in buona parte sugli utenti del servizio.
Basti pensare quindi a quanto è aumentata la stima del costo in termini monetari rispetto al 2001 per capire quanto si sta aggravando la situazione. Si stanno raggiungendo cifre spaventose e gli unici

che traggono profitto sono gli spammer. E' necessario correre ai ripari e porvi rimedio al più presto possibile.

2.5 SPAMMING PERMESSO, OPT-IN /OPT-OUT

In questi ultimi tempi l'aumentare a dismisura del fenomeno spam sta mettendo in crisi il core business del marketing che è da sempre rappresentato dalla pubblicità. Infatti fare pubblicità senza incorrere in sanzioni ormai è diventato sempre più difficile e questo risulta essere un grosso problema per tutte le aziende. Per addentrarci in questo campo prima di tutto bisogna chiarire il concetto di opt-in, a cui viene contrapposto l'opt-out.

Opt-in è un termine usato nel contesto del Web Marketing. Sottolinea l'obbligo del consenso preventivo del destinatario prima di inviare un messaggio email di stampo prevalentemente promozionale. Il destinatario, infatti deve averlo richiesto in precedenza attraverso un'iscrizione in un sito Web oppure attraverso una richiesta via email. Resta comunque sempre la possibilità di rimuoversi dal servizio. L'opt-in quindi sancisce il diritto alla scelta da parte dell'utente se ricevere o meno comunicazioni.

Opt-out al contrario indica un messaggio non richiesto di stampo prevalentemente promozionale che arriva nella casella di Posta Elettronica del destinatario. Permette comunque a chi lo riceve di rimuoversi dal servizio. Se così non fosse l'email ricevuta sarebbe sicuramente caratterizzata come spam. L'opt-out quindi lascia a posteriori all'utente la sola possibilità di essere escluso da ulteriori comunicazioni.

In questa materia inizialmente la scelta fra opt-in e opt-out era lasciata ai singoli stati. Se tutti adottassero la regola dell'opt-out sarebbe facile immaginare come ogni utente sarebbe bersagliato da email pubblicitarie ogni giorno vista la possibilità per le aziende e le persone fisiche di mandare almeno una comunicazione ad ogni indirizzo recuperato. Per questo molte organizzazioni mondiali, fra cui la CAUCE e molti siti internet si battono da anni per regolamentare lo schieramento sul fronte dell'opt-in. Purtroppo la scarsa regolamentazione in alcuni paesi (soprattutto asiatici) o l'aperta accettazione dell'opt-out di altri (come alcuni stati della federazione USA), rendono complesse ed a volte impossibili le battaglie contro lo spam. Il Parlamento Europeo poi ha approvato alcuni emendamenti che hanno stabilito che tutti i messaggi commerciali, inviati per email o sms, dovranno avvalersi del consenso preventivo.

Le basi legislative a cui è sottoposta l'attività di marketing on-line in Italia sono contenute nella legge 675/96 sulla privacy. La normativa prevede una serie di adempimenti burocratici da porre in essere sia nei confronti del garante per la protezione dei dati personali, sia nei confronti degli

interessati del trattamento, cioè coloro i cui dati vengono trattati. Per fare pubblicità in rete via email senza incorrere in sanzioni e senza passare per spammer (il cosiddetto spamming permesso) bisogna quindi seguire alcune precise regole:

1) gli indirizzi e-mail contengono dati personali, non sono pubblici, quindi la loro fruibilità su Internet non implica la facoltà di utilizzarli per l'invio di messaggi pubblicitari;
2) l'utilizzo di indirizzi ricavati da software di generazione automatica non è permesso se non con il consenso preventivo del destinatario;
3) è necessario richiedere il consenso del destinatario prima di inviare la mail e informarlo adeguatamente sugli scopi della mail. Vale quindi la regola dell'opt-in;
4) non è permesso l'invio di email senza indicare il mittente o la provenienza del messaggio o senza fornire indirizzi di recapito;
5) deve essere sempre garantita la possibilità per gli utenti di avvalersi dei diritti riconosciuti dalla normativa sulla privacy (ad esempio revoca del consenso, richiesta di conoscere la fonte dei dati, cancellazione dei dati dell'archivio, ecc.);
6) l'acquisto di banche dati contenenti indirizzi di posta costituisce l'obbligo di accertarsi se ciascuno dei presenti nella lista abbia manifestato il consenso preventivo per l'invio di materiale pubblicitario;
7) la sottoscrizione ad elenchi da parte di chi intende ricevere materiale pubblicitario o al contrario di chi non lo desidera non deve comportare oneri agli interessati;
8) se la fonte da cui si reperisce l'indirizzo è sottoposta ad un regime giuridico di piena conoscibilità da parte di chiunque, è invece possibile inviare e-mail pubblicitarie anche senza il preventivo consenso del titolare dell'indirizzo, con il solo obbligo di inviare l'informativa ex art.10, legge 675/96.

Ci sono standard ufficiali di rete, cui possa essere utile fare riferimento?

Qualcosa c'è: innanzitutto va citata la celebre RFC1855 dell'ottobre 95, nota come *netiquette,* in cui si viene avvertiti del problema e del fatto che l'unsolicited commercial email è considerata inaccettabile in rete, trattandosi di comunicazione "cost-shifted". Successivamente è stata varata la RFC2635, del giugno 99, che è specifica sul problema dello spam (che nel frattempo era divenuto molto più rilevante di quanto fosse nel 95). Essendo nella categoria *informational,* a rigore le suddette RFC non sono uno standard di rete, tuttavia inquadrano il problema in maniera ampia ed apprezzabile (soprattutto la 2635), costituendo quindi un riferimento che difficilmente qualcuno potrebbe ignorare (anche perché dietro tali RFC esiste in rete un consenso generale indiscutibile). Un'altra RFC interessante è la RFC2505, del febbraio 99, che contiene raccomandazioni antispam

assai importanti sul modo di gestire e configurare gli MTA. La RFC2505 è importante sia per gli amministratori di server che per gli sviluppatori del relativo software. Vi è poi la RFC3098 dell'aprile 2001, specificamente rivolta a chi intende esercitare attività di marketing in rete in maniera corretta. Si tratta di un documento assai utile e che va sicuramente letto. E' necessario mettere in guardia sul fatto che, nel trattare le mailing list, la RFC3098 illustra una modalità di gestione che non può essere considerata accettabile, in quanto mancante del doveroso passo di conferma della iscrizione da parte dell'utilizzatore della mailbox destinataria (operazione indispensabile per prevenire le iscrizioni fraudolente).

2.6 PROVVEDIMENTI LEGISLATIVI

Lo spam ai fini di profitto è un reato. Inviare email senza il consenso del destinatario è vietato dalla legge. Sono previste varie sanzioni e nei casi più gravi la reclusione.

Prima di giungere a questa definitiva conclusione però la strada si è rivelata molto lunga. I primi tempi si considerava lo spam non come un abuso ma come una necessità del mondo del marketing, solamente da limitare qualora si verificavano eccessi. Il primo Paese ad attivarsi per cambiare la tendenza sono stati gli USA. Qui infatti si costituì la CAUCE (Cohalition Against Unsoliticed Commercial Email), che sulla base della legge federale già in vigore che tutela l'abuso dei fax, doveva porre rimedio al vuoto legislativo in materia di email non richieste. Questo compito richiese molto tempo, anche per la continua controffensiva degli spammer che si attivarono per far legalizzare l'opt-out (ossia la possibilità di negare l'invio di email non richieste solo dopo averle ricevute). Nel 2003 finalmente il Congresso varò la nuova legge federale "CAN-SPAM Act of 2003". Tuttavia il problema rimase in quanto questa legge si fondò sul principio dell'opt-out e diede pieno controllo sulla pubblicità indesiderata solo ai provider, a scapito degli utenti che non potevano nemmeno agire in giudizio contro gli spammer.

Passando poi all'Europa si registrano anche qui vari tentativi per giungere ad una legislazione comune. Ne è una prova la "Direttiva 2002/58/CE del Parlamento Europeo e del Consiglio" del 12 luglio 2002, la quale costituì l'obbligo per gli Stati aderenti alla Comunità Europea di emanare provvedimenti legislativi sul principio dell'opt-in e quindi del preventivo consenso del destinatario.

Veniamo ora all'Italia. E' da notare che qualcosa in materia di email non richieste era già stato attuato seppur implicitamente. L'argomento ricade infatti nella legge sulla privacy, nota anche come legge 675/96 sulla protezione dei dati personali. In effetti l'indirizzo di posta elettronica è considerato come un dato personale anche se non contiene il nome del titolare. A scanso di

equivoci, in data 11 gennaio 2001, l'Autorità Garante per la protezione dei dati personali si è pronunciata in merito e ha stabilito l'illegittimità dell'utilizzo, senza preventivo consenso del titolare, di indirizzi ricavati da varie aree della rete, che anche se accessibili da chiunque non sono soggette ad alcun regime giuridico di piena conoscibilità.

La legge sulla privacy non vieta direttamente l'invio di posta commerciale ma limita l'uso dell'indirizzo di posta elettronica in determinati casi. Gli indirizzi e-mail reperibili in rete non sono pubblici e non possono essere usati per fini commerciali. Non basta quindi, per poter considerare pubblico un indirizzo di e-mail, il fatto che tale indirizzo sia conoscibile, in determinate circostanze, da una pluralità di persone come può succedere per un indirizzo pubblicato su Internet. Inoltre non possono essere considerati pubblici neanche gli indirizzi di email che vengono pubblicati su forum o newsgroup. Gli indirizzi e-mail in rete possono essere utilizzati solo per le finalità che hanno portato alla loro pubblicazione. Questo principio rende pertanto non conformi alla legge né la raccolta automatica di indirizzi di email presenti su internet né la loro creazione artificiosa, attività che si possono realizzare oggi con appositi software. Ma la legge non si ferma qui, infatti obbliga le persone fisiche o giuridiche a cui sono stati consegnati i dati, a fornire una descrizione chiara e precisa di quale uso ne verrà fatto: lettura, memorizzazione, trasferimento a terze parti, comunicazioni di servizio o comunicazioni commerciali; inoltre nel momento in cui si forniscono i dati, od in qualunque momento successivo, i titolari dei dati hanno il diritto di sapere entro 5 giorni dalla richiesta in quali termini verranno utilizzati od anche di limitarne o proibirne completamente l'uso.

Questo pronunciamento si rivelò di vitale importanza perché piegò la difesa degli spammer che si basava sulla classificazione degli indirizzi di posta elettronica reperiti sul web come pubblici e quindi consideravano la loro attività perfettamente consentita. E' possibile quindi perseguire contro gli spammer già grazie alla legge 675/96 anche se in realtà il procedimento si rivela lungo e costoso e soprattutto riguarda solo gli spammer italiani.

Sono state poi varate anche legislazioni più specifiche in materia. Per primo il decreto legislativo 171 del 1998 sancisce che il costo pubblicitario deve essere sostenuto interamente da chi fa la pubblicità e non da chi la subisce. Da segnalare anche il decreto legislativo n.185 del 22 maggio 1999 che, quando ancora la comunità europea non si era espressa in materia, schierò l'Italia sul fronte opt-in. In particolare nell'art. 10 impone limiti all'impiego di talune tecniche di comunicazione a distanza e tra questi dispone il preventivo consenso del destinatario per l'invio di posta elettronica non richiesta. Nell'art. 12 vengono invece indicate le sanzioni a cui incorre uno

spammer, ovvero a parte l'applicazione della legge penale qualora il fatto costituisca reato, chi ostacola il diritto del destinatario di recedere dall'invio di posta o il diritto al rimborso di eventuali spese, è punito con la sanzione amministrativa pecuniaria da lire un milione a lire dieci milioni. Inoltre in casi di particolare gravità o recidiva la multa viene raddoppiata.

Dopo una serie di interventi mirati alla sospensione di attività illecite o alla denuncia all'autorità giudiziaria di talune aziende o persone fisiche il Garante della privacy è sceso in campo in maniera chiara e dettagliata per disciplinare l'argomento. Il decreto legislativo 30 giugno 2003 n. 196, denominato "Codice in materia di protezione dei dati personali", entrato in vigore dall'1 gennaio 2004, infatti recepì nell'ordinamento italiano la direttiva europea 2002/58/CE e precisò vari aspetti legali riguardanti l'invio in internet di email promozionali o pubblicitarie. Si tratta del provvedimento legislativo principale a cui ci si può riferire in materia di UCE in Italia. In merito alle comunicazione indesiderate l'art. 130 sancisce il preventivo consenso, il legittimo invio di posta qualora sia direttamente l'interessato a fornire l'indirizzo fatta salva comunque la possibilità successiva di recedere dal consenso e il divieto di inviare posta celando o falsificando il mittente o non fornendo indirizzi di recapito. Per quanto riguarda i provvedimenti da attuare dà la possibilità ai provider di adoperare sistemi di filtraggio o altro in caso di reiterata violazione delle norme. L'art. 167 stabilisce le sanzioni penali: se dal fatto deriva nocumento, reclusione da sei a diciotto mesi o, se il fatto consiste nella comunicazione o diffusione, reclusione da sei a ventiquattro mesi. In definitiva quindi le sanzioni per chi viola le disposizioni di legge vanno dalla "multa", in particolare per omessa informativa all'utente (fino a 90mila euro); alla sanzione penale qualora l'uso illecito dei dati sia stato effettuato al fine di trarne per sé o per altri un profitto o per arrecare ad altri un danno (reclusione da 6 mesi a 3 anni). E' prevista anche la sanzione accessoria della pubblicazione della pronuncia penale di condanna o dell'ordinanza amministrativa di ingiunzione. Ulteriori conseguenze possono riguardare l'eventuale risarcimento del danno e le spese in controversia giudiziaria o amministrativa.

Per chi volesse avere maggiori dettagli e informazioni riporto nell'Appendice A il provvedimento del Garante della privacy.

3. TECNICHE CON CUI LO SPAMMER RECUPERA INDIRIZZI

I costi che uno spammer deve sostenere per la sua attività e per recuperare gli indirizzi sono veramente esigui. Infatti ricadono soprattutto sul destinatario. Questo non fa altro che incrementare e contribuire al dilagamento del fenomeno dello spam: la possibilità di farsi pubblicità in tutto il mondo contro un ridottissimo costo. Inoltre molti spammer usano spesso abbonamenti gratis. Gli spammer usano diverse tecniche per recuperare gli indirizzi che poi saranno metà dei loro loschi scopi. Fra queste troviamo:

- dictionary attack;
- liste di indirizzi;
- spambot;
- cavalli di troia;
- web bug;
- dialer.

3.1 DICTIONARY ATTACK

Questa tecnica molto diffusa si basa sull'indovinare gli indirizzi. Più precisamente lo spammer cerca di comporre e generare indirizzi che potrebbero effettivamente esistere. Per la parte destra della chiocciola @ usano nomi di dominio validi e per la parte sinistra compongono stringhe in base a qualche logica, per lo più nomi di persone.

Per questo motivo l'indirizzo nome.cognome@dominio.it è uno dei più soggetti a questo tipo di attacco. Se questi indirizzi effettivamente non esistono agli spammer non importa, comunque qualcuno riusciranno ad indovinarlo e questo già gratifica il loro sforzo di sparare nel mucchio.

3.2 LISTE DI INDIRIZZI

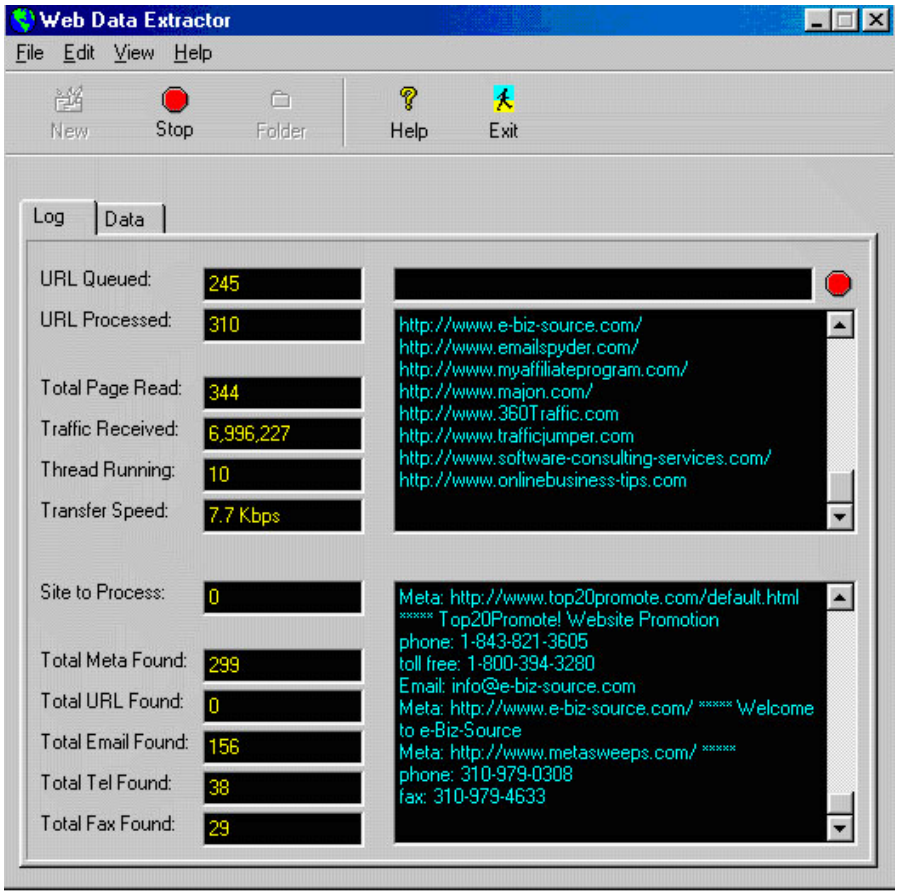
Un secondo sistema consiste nell'acquisire liste di indirizzi da soggetti che li raccolgono per poi rivenderli. Esiste un vero e proprio commercio su questo tipo di "affare".

3.3 SPAMBOT

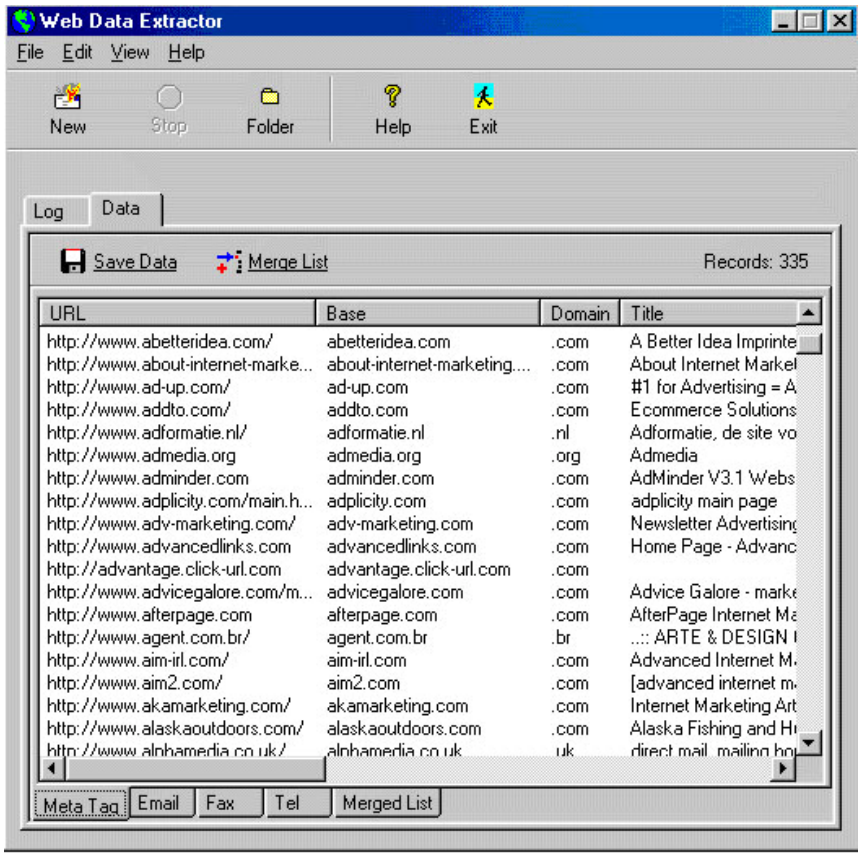
La fonte principale di indirizzi è rappresentata dai sistemi di mailing-list, gruppi di discussione che proprio si fondano sulle e-mail, e dal web. Per quanto riguarda le mailing-list ci si accede per richiedere l'iscrizione del proprio indirizzo per poi ricevere le notizie che riguardano quella

mailing-list. Per contrastare il problema dell'iscrizione di indirizzi non consentita è stata introdotta la conferma di iscrizione. Rimangono comunque le mailing-list che non prevedono la verifica e proprio queste sono il bersaglio degli spammer. Per quanto riguarda i newsgroup il campo più a rischio risulta essere `From:`.

Per questo lavoro gli spammer usano particolari software, chiamati spambot, costruiti a questo scopo: scandagliare il web alla ricerca di indirizzi. Si basano sullo stesso principio del funzionamento degli spider dei motori di ricerca ma a differenza di questi ultimi estraggono dalle pagine web tutti gli indirizzi presenti. In particolare lo spambot ricerca nel codice HTML della pagina il link mailto:utente@dominio.com oppure tutto ciò che contiene il carattere @. Ne esistono diversi tipi, possono essere fatti in casa oppure ci sono aziende che li vendono a volte anche a prezzi molto elevati. Fra gli spambot commerciali più importanti troviamo Web Data Extractor (di cui vediamo due figure) che oltre ad indirizzi email estrae fax, numeri di telefono, URL, testo e tag particolari. E' disponibile al sito http://www.webextractor.com.



[Screen Shot - Main Window]



[Screen Shot - Data Window]

3.4 CAVALLI DI TROIA

Non è raro il fatto che gli spammer si appoggino a Cavalli di Troia che impiantano nel computer di utenti del tutto ignari di ciò che sta succedendo. In pratica questo tipo di minaccia informatica è un programma che viene eseguito all'insaputa dell'utente del computer su cui è stato impiantato e oltre ad agire come mailserver, recupera tutta la rubrica di posta del computer infettato. In questo modo il computer della vittima diventa origine di spam nascondendo il reale indirizzo dello spammer che purtroppo risulta impossibile da rintracciare.

3.5 WEB BUG

Un altro trucchetto molto utilizzato dagli spammer è quello di scrivere lo spam sottoforma di messaggi formattati in HTML e in questo modo inserire immagini nascoste con precise funzioni di dimensioni 1x1 pixels, in modo che sia invisibile. Gli spammer fanno affidamento sul fatto che molti mailreader traducono il codice HTML con le funzionalità del browser e facendo questo attivano gli script presenti. L'immagine viene scaricata e avvisa lo spammer che la persona che ha ricevuto lo spam legge la posta e che l'indirizzo risulta effettivamente attivo.

Un'altra insidia che porta questa immagine è rappresentata dai cookie. Infatti allo scaricamento di essa verrà spedito un cookie al server dello spammer contenente informazioni che potrebbero essere

associate all'indirizzo e-mail al quale lo spam è stato inviato in modo da rendere possibile la costruzione di un archivio di corrispondenze fra cookie e indirizzi di posta molto utile per lo spammer. In questo modo infatti gli spammer riescono a ricavare l'indirizzo e-mail e ad incrementare il loro giro di affari.

Non solo, esiste la concreta possibilità che possano creare un archivio completo degli argomenti di interesse di ogni sfortunato utente che gli capiti sottomano. Infatti non è raro che in alcuni siti siano presenti dei banner o delle immagini nascoste contenenti il sito dello spammer e questo dà l'opportunità di poter recuperare una sorta di storia dei siti visitati dagli utenti per avere una panoramica delle loro aree di interesse e quindi veicolare lo spam in base ad esse.

Questa tecnica dell'immagine nascosta e del cookie viene chiamata web bug. E' necessario comunque precisare che non tutte le immagini di dimensioni 1x1 pixels siano da considerarsi web bug, a volte vengono usate solo per la formattazione della pagina e non è detto che questo monitoraggio sia per cattivi scopi. Resta comunque il fatto che bisogna diffidare da essi e prestare molta attenzione per non cadere nella trappola.

3.6 DIALER

Una minaccia sempre più consistente riconducibile allo spam è l'avvento dei programmi dialer. In pratica il destinatario viene invitato ad eseguire questo programma (rappresentato da un file .exe) che prende il controllo del modem ed effettua una chiamata attraverso la linea telefonica dell'utente. Il problema è che si accede ad una rete privata che offre determinati prodotti però la tariffa di connessione non è la solita. Si tratta di numerazioni molto costose, chiamate anche a valore o a costo aggiunto, spesso riconoscibili dal prefisso (ad esempio 709 o 899). A volte il dialer viene anche spedito in allegato allo spam senza necessità di cliccare su alcun sito. A parte la denuncia purtroppo non esistono molte soluzioni a questo problema.

4. COSA FARE QUANDO SI RICEVE SPAM

La prima cosa da ricordare quando si riceve spam è assolutamente non rispondere allo spammer, qualunque sia il contenuto della mail. Infatti la risposta per lo spammer ha una unica finalità: la conferma che l'indirizzo è esistente, valido, attivo e per di più usato. Quindi l'indirizzo assume più valore anche commerciale; lo spammer può infatti benissimo inserirlo in altre liste ma soprattutto può rivenderlo ad altri spammer. Poi non è detto che ci sia effettivamente un indirizzo a cui rispondere nell'e-mail oppure frequentemente, per sviare i sospetti, sono indirizzi non validi o appartenenti a qualcuno che non c'entra nulla.

Sarebbe meglio anche non aprire le e-mail di spam. Infatti potrebbero contenere link a immagini fittizie che vengono caricate dal sito web dello spammer e grazie a queste riescono a verificare che il loro messaggio è stato aperto e quindi il vostro indirizzo di posta è valido e attivo. Per lo stesso identico motivo è buona norma disattivare l'eventuale anteprima fornita dal vostro client di posta.

E' importante anche non seguire le istruzioni che a volte vengono date nel messaggio specialmente per quelle che offrono di rimuovere l'indirizzo dalla propria lista e quindi promettono che non si riceverà più posta di quel tipo da loro. Prima di tutto non è garantito che effettivamente rispettino i patti ma anzi più facilmente in questo modo si viene inseriti in altre liste.

L'opt-out ovvero la possibilità di recedere dalla ricezione di e-mail solo dopo il primo invio e con una richiesta esplicita non è da considerarsi corretto perché toglie il diritto di scelta all'utente e come già detto l'Europa si è schierata sul fronte dell'opt-in, ovvero il consenso preventivo.

Inoltre non si deve agire con gli stessi mezzi dello spammer per vendicarsi con ritorsioni tipo mailbombing. Questa mossa sarebbe solo controproducente e potrebbe danneggiare altri utenti.

Non resta che adoperarsi sfruttando mezzi leciti e soprattutto più efficaci. Per difendersi si può agire su quattro fronti: *prevenzione*, *denuncia*, *bloccaggio* e *filtraggio*.

Il primo aspetto consiste nell'applicare e seguire una certa serie di regole e trucchetti per evitare che gli spammer entrino in possesso del nostro indirizzo di posta elettronica.

Per provvedere a posteriori invece ci viene in aiuto la possibilità di denunciare gli spammer in due modi: attraverso reclami e segnalazioni direttamente ai provider che forniscono connettività e possono toglierla o ad organi competenti della Abuse oppure in casi estremi attraverso la denuncia al Garante per la protezione dei dati personali. Si tratta quindi di un'azione per cercare di eliminare alla fonte il problema anche se ciò può richiedere del tempo.

Per quanto riguarda il bloccaggio ci si riferisce a liste di blocco che contengono vari indirizzi IP di risorse fonti di spam. Questa tecnica consiste, quindi, nel rifiutare tutta la posta proveniente da indirizzi appartenenti a quella lista.

Un’ultima importante arma è costituita dai filtri da applicare ad un client di posta elettronica oppure ad un programma che analizza le email in ingresso e che in base a certe regole, anche impostare dall’utente, dia la possibilità di separare le e-mail legittime da quelle spazzatura.

5. REGOLE E TRUCCHI PER LA PREVENZIONE

Il bene più prezioso per gli spammer è rappresentato dagli indirizzi e-mail, quindi la prima difesa principale da attuare contro questo tipo di minaccia consiste nel rendere difficile il reperimento del proprio indirizzo di posta elettronica.

5.1 CONSIGLI UTILI

Esistono vari accorgimenti o trucchetti che si possono adottare per evitare di cadere nelle mani degli spammer. Vediamone alcuni:

- Limitare la diffusione del proprio indirizzo email sul Web, ovvero non diffondere il proprio indirizzo con leggerezza soprattutto quando ci si iscrive a determinati siti oppure aprire indirizzi adibiti allo scopo per evitare rischi.
- Utilizzare la Ccn (Carbon copy nascosta) o Bcc (Blank Carbon Copy). Si tratta di inserire gli indirizzi di destinatari che riceveranno una copia del messaggio, ma della cui esistenza tutti gli altri destinatari saranno ignari.
- Non diffondere i "virus alerts", ovvero i messaggi che avvisano della nascita e diffusione nuovi virus. Prima di tutto possono considerarsi anche questi dei messaggi di spam e inoltre non è il modo migliore per contrastare la diffusione dei virus.
- Mascherare il proprio indirizzo quando si partecipa ad un gruppo di discussione (newsgroup). Gli spammer infatti fanno uso di particolari programmi detti "bot" che recuperano direttamente dai newsgroup tutti gli indirizzi possibili.
- Utilizzare programmi di filtraggio automatico.
- Non rispondere mai ad un messaggio di spam e soprattutto non farlo in toni offensivi. Si potrebbe venire incontro a mail bomb (bombardamento di email) o flame war (guerre di insulti) e soprattutto è il modo migliore per confermare allo spammer che l'indirizzo esiste ed è utilizzato.
- Non accettare mai l'invito a rimuovere il proprio nominativo dalla loro lista. Inoltrando la richiesta di cancellazione lo spammer riceverebbe conferma che l'indirizzo a cui ha inviato la mail sia un indirizzo esistente e valido.
- Disattivare la visualizzazione dell'anteprima nei propri client di posta. Questa tecnica serve ad impedire che un'e-mail in HTML contenente del codice sia in grado di inviare il vostro indirizzo e-mail al mittente e inoltre impedisce la visualizzazione di immagini non adatte a minori.

- Interrompere le cosiddette catene di Sant'Antonio. Quasi sempre queste storie stucchevoli e piene di dolore sono create al fine di ottenere una risposta ed accrescere il database di indirizzi dello spammer.
- Non tenere in evidenza l'indirizzo email neanche nelle chat.
- Non mettere il proprio indirizzo in siti web. Gli spammer hanno programmi di scansione anche per le pagine web. I motori di ricerca degli spammer in questo caso oltre a cercare indirizzi nel testo della pagina provvedono a individuare tutti i link con una voce *mailto:* nel campo *href*. I problemi quindi sono due: mascherare l'indirizzo visualizzato e mascherare il contenuto del link relativo. Una semplice possibilità per nascondere il testo è quella di creare un'immagine contenente l'indirizzo email. Nessuno spider sarà mai in grado di individuarne il contenuto. Il mascheramento del link invece può essere realizzato tramite javascript che codifichino l'intera stringa *href* con gli equivalenti codici ASCII e che scrivano il risultato sulla pagina in modo dinamico.
- Evitare di fornire l'indirizzo email compilando dei form sui siti web.
- Scegliere indirizzi di posta non troppo brevi e non banali. Sconsigliato l'uso di *nome.cognome* prima del dominio. Gli spammer utilizzano appositi software per creare automaticamente combinazioni di indirizzi di posta.
- Utilizzare servizi di forwarding gratuito per generare alias con eventuali descrizioni (es. questo indirizzo l'ho usato per iscrivermi al sito xxx, ecc.); Questa tecnica si rivela particolarmente utile per poter gestire un differente indirizzo per differenti occasioni e in particolare consente di stabilire chi ha venduto l'indirizzo allo spammer qualora iniziassero a manifestarsi i primi casi.
- Evitare di far inserire il proprio indirizzo di posta su servizi di elenchi online, sono le fonti preferite dagli spammer. Molti fornitori di servizi di posta offrono questa possibilità all'atto dell'iscrizione. Verificate che questa eventuale opzione non sia selezionata in modo predefinito.
- Leggere sempre attentamente le clausole dei contratti di un fornitore di un qualsiasi servizio online che prevede l'iscrizione tramite un indirizzo email. Spesso è previsto che l'indirizzo possa essere ceduto a terzi per l'invio di materiale informativo. Se non è possibile evitare questa possibilità si valuti la scelta di un servizio diverso.
- Si valuti la possibilità di utilizzare servizi di alias di posta che includono il filtraggio antispam. Un ottimo servizio gratuito è offerto da http://www.despammed.com. Il servizio non fornisce una nuova casella di posta ma semplicemente un nuovo alias del tipo *nomeutente@despammed.com* che redigerà la posta nella vostra reale casella solo dopo

averla ripulita dallo spamming. Questa scelta si rivela utile tutte le volte che ci è indispensabile pubblicare sulla rete un indirizzo al quale farci raggiungere.

5.2 ADDRESS MUNGING

Una tecnica molto utile per evitare la raccolta di indirizzi risulta essere il mascheramento e la falsificazione degli indirizzi di posta. Questo sistema è detto *address munging*, che deriva dalla parola munge che significa rompere. Si tratta di alterare l'indirizzo in modo tale che gli esseri umani possano riconoscerlo, ma che ciò non sia possibile per i software degli spammer. E' necessario operare in vari modi:

- Fare in modo che la parte dopo la @ diventi un riferimento non valido;
- Rendere l'alterazione il più possibile evidente
- Consigliabile far terminare l'indirizzo con .invalid
- Inserire comunque indicazioni che consentano di ricavare l'indirizzo reale

Vediamo alcuni esempi:

utente@dominio.it può diventare utente@dominioNOSPAM.it oppure utente@TOGLIMIdominio.it o utenteTOGLIQUESTO@dominioTOGLIANCHEQUESTO.it.

Queste parole sono considerate ormai classiche e prevedibili, come anche REMOVE, REMOVE.THIS.TO.REPLY. Infatti gli spammer hanno aggiornato i loro software in modo da ricercare ed eliminare le stringhe più comuni. Per questo è necessario usare parole più creative e cambiarle di tanto in tanto; l'importante resta sempre dare una spiegazione di come recuperare l'effettivo indirizzo.

Altri esempi sono la sostituzione della @ con "AT" e del punto con "DOT".

Per proteggere l'indirizzo senza rinunciare al mailto e senza obbligare l'utente a cancellare le parole civetta sono state introdotte alcune tecniche:

- Usare javascript
- Codificare l'indirizzo
- Hiveware Enkoder Form

Vediamole tutte più in dettaglio.

5.2.1 Usare Javascript

E' certamente il metodo più diffuso. Permette di inserire l'indirizzo email nel sito internet e in particolare nel punto in cui vogliamo venga visualizzato l'indirizzo è necessario aggiungere questo codice:

```
<script language="JavaScript">
<!--
var nomeutente="utente";
var dominio="nomedominio.com";
var testo="Scrivimi";
document.write("<a href=" + "mail" + "to:" + nomeutente +
 "@" + dominio + ">" + testo + "</a>") ;
//-->
</script>
```

Per adattarlo alle proprie esigenze basterà personalizzare i valori delle variabili `nomeutente`, `dominio` e `testo`, che rappresenta il testo che compare nella pagina per poter mandare un'e-mail. In pratica questo script ha il compito di confondere il software dello spammer dividendo in più parti l'indirizzo e l'espressione mailto. L'unico inconveniente è rappresentato dal fatto che questa tecnica non funziona con browser che non supportano o hanno disabilitato il JavaScript.

Sun Star Media ha diffuso una variante di questo script, la quale va inserita nella sezione <head> della pagina e poi verrà richiamata successivamente passando alcuni parametri.

```
<script language="JavaScript">

<!--
function DisplayMail(Server, Login, Display){
if ((Display.length == 0) || (Display.indexOf('@')+1)) {
document.write("<a href=" + "'mai" + "lto:" + Login + "@" + Server + "'>" +
Login + "@" + Server + "</a>"); }
else {
document.write("<a href=" + "'mai" + "lto:" + Login + "@" + Server + "'>" +
Display + "</a>"); }
}
//-->
</script>
```

Quindi per richiamare la funzione, nel punto dove vogliamo collocare l'indirizzo, basterà inserire questa riga di codice:

```
<script language="javascript">DisplayMail('dominio.com', 'utente', '');</script>
```

Lasciando vuota la terza variabile, verrà visualizzato nella pagina l'indirizzo. Altrimenti, possiamo inserire un testo adatto allo scopo in questo modo:

```
<script language="javascript">DisplayMail('dominio.com', 'utente',
'Scrivimi');</script>
```

5.2.2 Codificare l'indirizzo

Un'altra possibilità è quella codificare l'indirizzo con i valori delle corrispondenti entità HTML. Quindi per esempio l'indirizzo utente@dominio.com risulta essere:

```
&#117;&#116;&#101;&#110;&#116;&#101;&#064;&#100;&#111;&#109;
&#105;&#110;&#105;&#111;&#046;&#099;&#111;&#109;
```

In questo modo il browser visualizzerà l'indirizzo correttamente mentre il software dello spammer non sarà in grado di tradurlo. Effettuare questa codifica non sembra del tutto semplice, per questo motivo esistono alcuni servizi online che permettono di ottenere la codifica semplicemente inserendo solo l'indirizzo.

Fra questi i più importanti sono Email Encoder (http://www.wbwip.com/wbw/emailencoder.html) e SiteUp Networks (http://www.siteup.com/encoder.html), dove il codice completo (link + codifica) arriva via mail.

L'unica cosa che resta da fare è quindi copiare il codice nel link malto: così:

```
<a href="mailto:&#117;&#116;&#101;&#110;&#116;&#101;&#064;
&#100;&#111;&#109;&#105;&#110;&#105;&#111;
&#046;&#099;&#111;&#109;">Scrivimi</a>
```

Tuttavia questo metodo non si rivela infallibile: anche se non presenta problemi di supporto poiché non si tratta di uno script, resta pur sempre la possibilità che gli spammer aggiornino i loro software in modo tale da renderli in grado di decodificare il codice.

5.2.3 Hiveware Enkoder Form

Si tratta della tecnica più sicura fra quelle proposte. Si basa su un tool, Form Enkoder (http://www.hiveware.com/enkoder_form.php), creato da Dan Banjamin, che si compone di una combinazione di JavaScript e codifica di caratteri. Nel Basic Form devono essere inseriti:

- l'indirizzo da codificare;

- il testo da far comparire al posto dell'indirizzo;
- il testo per l'attributo title che compare come tooltip passandoci sopra con il mouse;
- ed eventualmente l'oggetto della mail.

Dopo si procede come per le altre tecniche copiando il codice nel link mailto.

Queste soluzioni si sono rivelate abbastanza efficaci ma riguardano solo il lato client. Per il lato server occorrono tecniche di redirect e riscrittura degli URL.

Alcuni webmaster hanno iniziato ad utilizzare programmi CGI, come WebPoison, programma gratuito che genera pagine contenenti testo random, indirizzi e-mail fasulli, e link che puntano a pagine analoghe. Tutto ciò per fare in modo che i software degli spammer girino all'infinito tra queste pagine recuperando solo indirizzi falsi. Tuttavia questa tecnica di certo non elimina il problema, ma è utile per infastidire gli spammer.

6. COME RICAVARE L'INDIRIZZO DELLO SPAMMER

Per poter avviare una controffensiva contro gli spammer prima di tutto è necessario individuare la provenienza delle e-mail ricevute e per farlo si devono analizzare gli header. E' necessaria una precisazione: è il mail client che si occupa di mettere le intestazioni al messaggio di posta da trasmettere al server SMTP. Il mail client è sotto il controllo del mittente che decide quale adottare ed eventualmente come configurarlo. Quindi lo spammer utilizzerà programmi adatti alle sue esigenze e soprattutto fornirà header non attendibili e fasulle per simulare mail client tradizionali e soprattutto per trarre in inganno chi cercasse di individuarlo. Tuttavia lo spammer potrebbe anche fare a meno del client e collegarsi al server mediante telnet procedendo a mano.

6.1 ANALISI DELLE INTESTAZIONI

Innanzitutto il campo `From:` non è da considerarsi autentico: lo spammer non avrà nessun interesse a indicare quello corretto, anzi deve metterlo falso altrimenti sarà facilmente individuabile. Nemmeno il campo `To:` ha valore, infatti non è in base al suo contenuto che si determina a chi è destinato il messaggio. Infatti è il campo `Rcpt To:` quello a cui si deve far riferimento per scoprire il destinatario. Lo spammer farà in modo che al server vengano dati qualche centinaia o migliaia di comandi `Rcpt To`: ciascuno con l'indirizzo di una vittima che riceverà una copia del messaggio di spam.

Per riuscire ad individuare lo spammer l'unica header che ci può aiutare è `Received:`. Infatti questa intestazione, aggiunta dal server, permette di testimoniare che il messaggio è transitato di lì e da chi è stato ricevuto. Vediamo un esempio:

```
Received: from mariorossi (ppp26-milano.prima-rete-esempio.com [194.188.15.26])
          by mail.prima-rete-esempio.com (8.8.5/8.8.5) with SMTP id RU387493
      for <fverdi@seconda-rete-esempio.com>; Wed, 15 Apr 1998 14:26:32 +0200
      (METDST)
```

Il nome del server che ha inserito l'intestazione viene dichiarato dopo la parola `by.` In pratica l'header va letto in questo modo: "Questo messaggio è stato ricevuto su mail.prima-rete-esempio.com, proveniente da qualcuno che si è presentato come mariorossi e che comunque aveva l'indirizzo IP 194.188.15.26. Tale indirizzo risulta corrispondere alla risorsa di nome ppp26-milano.prima-rete-esempio.com". Dopo la parola `from` c'è il nome con cui si è presentato tramite il comando HELO chi ha passato il messaggio al server e in caso di spam questo è sicuramente fasullo. Quindi ciò su cui ci deve basare è l'indirizzo IP che compare tra parentesi quadre e il nome della risorsa che corrispondono alla provenienza del messaggio.

Altre informazioni che si possono ricavare dall'esempio sono la versione del software installato sul server (8.8.5/8.8.5), il numero di serie assegnato alla transazione indicato da `SMTP id`, infine l'indicazione del destinatario dopo la parola `for` e la data e l'ora.
Sempre facendo riferimento all'esempio si può notare che il dominio del server destinatario è `seconda-rete-esempio.com` e quindi una volta interrogato il server DNS (di cui parleremo fra breve) riesce ad individuare il nome del server a cui passare il messaggio in questo caso `plutus.seconda-rete-esempio.com`. Quest'ultimo inserirà una header `Received:` in testa e così via.

```
Received:      from      prima-rete-esempio.com      (mail.prima-rete-esempio.com
[194.184.145.216])
         by plutus.seconda-rete-esempio.com (8.8.5/8.8.5) with SMTP id GC502624
         for <fverdi@seconda-rete-esempio.com>; Wed, 15 Apr 1998 14:27:15 +0200
(METDST)
Received: from mariorossi (ppp26-milano.prima-rete-esempio.com [194.188.15.26])
         by mail.prima-rete-esempio.com (8.8.5/8.8.5) with SMTP id RU387493
         for <fverdi@seconda-rete-esempio.com>; Wed, 15 Apr 1998 14:26:32 +0200
(METDST)
From: mario.rossi@prima-rete-esempio.com (Mario Rossi)
To: fverdi@seconda-rete-esempio.com
Date: Wed, 15 Apr 1998 14:24:06 +0200
Message-ID: <3754F6E5.BA67D052@prima-rete-esempio.com>
X-Mailer: Gorilla 3.2 (Win95; I)
Subject: Richiesta informazioni sulle vs. iniziative
```

Quindi una volta visualizzati gli header si deve passare ad analizzarli e si parte proprio da `Received:` per verificare quali sono autentici e si inizia a percorrere la catena a ritroso dal ricevente al mittente che dovrebbe essere l'ultimo dall'alto al basso salvo l'inserimento di `Received:` ingannevoli. E' doveroso tener anche presente che alcuni spammer sono anche provider di sé stessi.
Ad ogni modo per ogni `Received:` è necessario verificare che sia confermato dall'header precedente nel messaggio ma successivo nel tempo. In particolare se il server che ha aggiunto l'intestazione e che è indicato dopo `by` è quello indicato nell'header precedente come provenienza. Inoltre si dovrà testare l'effettiva corrispondenza fra nome risorsa e indirizzo IP.
Il sistema che si occupa di queste corrispondenze e che gestisce questa immensa tabella di conversione è il Domain Name Server ovvero il DNS. Esso si avvale di un database distribuito con la rete suddivisa in zone servite da un DNS competente solo per le risorse appartenenti alla sua

zona. Quindi per risolvere le corrispondenze si dovrà contattare il server DNS passandogli il nome del dominio, ad esempio EXAMPLE.COM. Questo verrà spezzato in modo da estrapolare la parte più a destra detta top-level domain (.COM), che rappresenta il dominio di massimo livello che avrà un server DNS competente. Quindi la richiesta verrà passata al DNS competente per .COM che cercherà a sua volta il server competente per EXAMPLE.COM ottenendo l'indirizzo IP richiesto. Esiste anche il reverse DNS che si occupa del lavoro inverso, ovvero da un indirizzo IP ricavare il nome di dominio corrispondente, anche se non è molto sicuro: a volte non produce risultati, altre fornisce risultati falsificati. Questo perché si sono avuti casi di reti dedicate ad ospitare spammer.

6.1.1 Database WHOIS on-line

Per verificare le corrispondenze ci vengono in aiuto i database WHOIS on-line. Si tratta di sistemi che contengono tutti i record relativi alle assegnazioni di indirizzi di rete e nomi di dominio resi pubblici e consultabili da chiunque. Si possono reperire varie informazioni ulteriori come ad esempio indirizzi e contatti dei responsabili.

Per effettuare interrogazioni a questi database si possono usare appositi client oppure più semplicemente consultarli tramite un browser web. Per l'area del Nord America e in generale per quanto riguarda i più diffusi domini di massimo livello (.com,.net,.org) si può far riferimento a questi link:

- http://www.arin.net/whois/index.html per capire a chi è assegnato un indirizzo di rete;
- http://www.netsol.com/cgi-bin/whois/whois per capire a chi è assegnato un certo dominio.

Ad ogni modo per capire a chi fare riferimento basti ricordare che l'assegnazione degli indirizzi è gestita dalla IANA (Internet Assigned Numbers Authority, www.iana.org) che delega i compiti a vari organismi per macroaree geografiche. In particolare:

- ARIN (www.arin.net) per il Nord America;
- LACNIC (www.lacnic.net) per l'America del Sud e Centrale;
- RIPE (www.ripe.net) per L'Europa (compresi Medio Oriente e alcuni paesi dell'Asia);
- APNIC (www.apnic.net) per l'Asia e l'area del Pacifico;
- AFRINIC (www.afrinic.net) per l'Africa.

Esiste un trucchetto per scoprire eventuali Received fasulli; infatti se si trovasse un nodo dial-up sarebbe necessariamente quello la fonte di origine del messaggio.

Se si ha disposizione l'indirizzo IP si può identificare la rete alla quale quell'indirizzo è stato assegnato e quindi anche l'indirizzo, il telefono e i nomi dei responsabili. Se però si tratta di indirizzi dinamici non si riesce ad ottenere queste informazioni, ma il provider certamente ha i

mezzi per individuare l'utente. Se invece si possiede l'indirizzo email dalla parte significativa, ovvero il nome del dominio dopo la chioccola, si può arrivare al server.
In ogni caso è necessario tener presente che non sempre è chiarissimo stabilire chi è il responsabile dello spam, un'indicazione significativa si può comunque ottenere individuando chi è il responsabile per il reverse DNS sull'indirizzo IP in questione.

6.1.2 Sender Policy Framework

Il Sender Policy Framework (SPF) si propone di rendere più difficoltosa la falsificazione del campo `Return-Path:` al fine di identificare meglio gli spammer. Il protocollo SMTP non garantisce che il mittente del messaggio sia effettivamente chi dichiara di essere. L'MTA mittente deve solo passare un comando valido tramite il comando `MAIL FROM:`, il quale viene utilizzato dal ricevente per stabilire il `Return-path:`. Come abbiamo già visto, il `Return-Path:` non coincide sempre con il campo `From:`.
SPF si propone di salvare nella zona del DNS di ciascun dominio gli indirizzi IP dei server SMTP dai quali gli utenti che hanno un account su quel dominio solitamente inviano la posta. Ad esempio se gli utenti del dominio `@miodominio.it` inviano posta solo da due server SMTP con indirizzi IP 1.2.3.4 e 1.2.3.5, occorrerà creare un record `TXT` per l'hostname `miodominio.it`:

```
miodominio.it IN TXT "v=spf1 ip4:1.2.3.4 ip4:1.2.3.5"
```

Quindi ogni volta che un server SMTP con supporto per SPF riceve una mail, controlla se il dominio fornito dal mittente ha un record che inizia per `v=spf1`. Se c'è vengono confrontati gli indirizzi IP e in caso questo controllo non abbia successo (non combaciano gli indirizzi) è altamente probabile che il dominio del mittente sia falsificato. E' possibile creare automaticamente il `TXT` record per i propri domini utilizzando l'apposito wizard: http://spf.pobox.com/wizard.html.

6.2 ANALISI DEI SITI DELLO SPAMMER

Nelle e-mail contenenti spam a volte sono presenti siti inseriti dallo spammer per pubblicizzare meglio la loro attività.

6.2.1 Siti click-through

Spesso il sito che viene pubblicizzato nello spam non è il vero sito dello spammer ma contiene un riferimento ovvero un link per raggiungerlo. Quindi il primo sito viene chiamato click-through in quanto se ci si clicca su in qualche modo si viene reindirizzati al sito dello spammer. Il sito può essere raggiunto direttamente o seguendo una sequenza di pagine magari anche altri siti click-through. Molto spesso si avvalgono di form per recuperare gli indirizzi di posta.

E’ utile accedere questo siti non con il proprio browser ma con il safe browser disponibile online al sito http://www.samspade.org/. Con questo non solo viene presentato il sorgente della pagina web ma anche gli header inviati dal server, ovvero ciò che restituirebbe il server come risposta ad un comando HTTP GET. In questo modo si riesce a stabilire facilmente dove sono ubicati i siti degli spammer e si può arrivare a questi senza far scoprire la propria provenienza e senza accettare.

6.2.2 Siti nascosti in routine Javascript

A volte gli spammer nascondono il link al loro sito in una procedura javascript. In pratica quando viene caricata la pagina su un browser, se questo è in grado di decifrare il javascript la routine verrà eseguita e produrrà righe di codice html che reindirezzeranno il browser verso la pagina web dello spammer. Spesso queste procedure sono fornite da programmatori che vendono spamware, ovvero software di supporto all’attività dello spammer. Una volta decifrata la routine il browser visualizzerà la pagina contenuta in essa senza dar la possibilità di poter vedere il codice html, quindi per poter risalire ad esso è necessario alterare il sorgente originario. Ciò si può fare in diversi modi:

1. Una prima soluzione potrebbe essere comprendere lo script fra il tag <XMP> e il tag </XMP> che dice al browser di visualizzare tutto ciò che è compreso fra questi due tag cosi com’è. Anche se deprecato questo tag risulta ancora supportato da molti browser.
2. Seconda soluzione consiste nel sostituire i caratteri “<” “>” con “&lt” e “&gt” in modo da renderli normali caratteri di testo. Risulta necessario comprendere lo script nei tag <PRE> e </PRE> per ottenere l’originaria formattazione. Il vantaggio risiede nel fatto che una volta salvato questo file può essere visto da tutti i browser.
3. La terza soluzione si basa sull’aggiunta di istruzioni in modo tale che il codice decodificato contenente l’URL del sito dello spammer venga visualizzato come input in una form e più precisamente in una TEXTAREA.

7. TECNICHE DI FALSIFICAZIONE E OCCULTAMENTO DELL'INDIRIZZO IP

Sempre più spammer si stanno adoperando per affinare tecniche per falsificare il proprio indirizzo IP o per nasconderlo e fare in modo di non essere scoperti per continuare la loro losca attività indisturbati. Vista la proibizione della loro pratica da parte degli ISP, gli spammer si stanno dirigendo verso sistemi vulnerabili come gli open mail relay e server proxy aperti. Non da meno è l'abuso nei confronti dei remailer anonimi, che usano sistemi di crittografia per rendere estremamente difficile individuare l'identità reale del mittente di un messaggio di posa elettronica. L'unico risultato è quello di negare la loro utilità agli utenti legittimi, che si vedono disabilitare queste risorse.
Esistono vari trucchi per falsificare e occultare l'indirizzo IP, ne esaminiamo i più importanti.

7.1 FALSIFICAZIONE DEGLI HEADER "RECEIVED"

Qui va fatta subito una precisazione: lo spammer può sì falsificare gli header `Received` ma non tutti, solo gli ultimi della catena. Infatti non può evitare che i server che trasportano il messaggio lascino una prova dell'avvenuto passaggio aggiungendo la propria header `Received`. Per falsificare l'header lo spammer può inserire una stringa arbitraria tramite HELO e quindi nella casula `from` della header `Received` viene visualizzata questa stringa priva di significato. Può inserire inoltre indirizzi IP che non c'entrano nulla e anche stringhe talmente lunghe che impediscono l'apparizione delle parti significative nelle header. Per esempio potrebbe inserire anche ulteriori clausole `from`, `by`, `with SMTP id` fasulle che potrebbero confondere anche se si può risalire facendo sempre attenzione al fatto che quella da considerare giusta è l'ultima come si può vedere nell'esempio dove la parte in grassetto è quella che va tenuta in considerazione.

```
Received: from myrop (ew6.southwind.net [216.53.98.70]) by
onyx.southwind.netfrom homepage.com (114.230.197.216) by
newmail.spectraweb.ch from default (m202.2-25.warwick.net [218.242.202.80])
byhost.warwick.net (8.10.0.Beta10/8.10.0.Beta10) with SMTP id e9GKEKk19201
([63.44.155.8]) by servidor2.bauer.es (Lotus SMTP MTA v4.6.1  (569.2
2-6-1998)) with SMTP id C1256A17.00129C3A; Sun, 22 Mar 1970 04:23:31 +0100
```

Risulta comunque difficile che l'indirizzo IP che troviamo nel header `Received` sia falsificato, più frequentemente viene utilizzato un altro indirizzo che nulla ha a che fare come vedremo dopo.

7.2 REMAILER ANONIMO

In alcuni casi non si riesce a stabilire il punto in cui il messaggio è stato messo in rete perché l'header inserito da un server non dice da chi ha ricevuto il messaggio. Quindi purtroppo non si riesce a risalire allo spammer.

Un server, infatti, non è tenuto a dichiarare chi è stato a passargli il messaggio e quindi non inserisce la clausola `by` dopo il `from` nella header `Received`. Quando questo avviene, siamo in presenza o di un *anonymous remailer* o di un server gestito da personale non competente. Sono proprio questi i bersagli preferiti dagli spammer che utilizzano questo tipo di server per veicolare il loro traffico senza essere scoperti.

Il funzionamento alla base dei remailer è semplice: ogni remailer elimina gli header e re-invia la mail. Esistono tre tipi di remailer:

1. Remailer di tipo 0, Pseudo-Anonymous;
2. Remailer di tipo 1, Cypherpunk;
3. Remailer di tipo2, Mixmaster.

7.2.1 Remailer di tipo 0, Pseudo-Anonymous

Questi furono i primissimi remailer in circolazione. Consentivano di inviare e ricevere e-mail in maniera estremamente semplice: si inviava la mail al server e sulla prima riga si scriveva il destinatario del messaggio, preceduto dal comando *"X-Anon-To:"*. Il server rimuoveva l'header dalla mail, sostituiva l'indirizzo con un ID generato casualmente e lo inviava al ricevente.

7.2.2 Remailer di tipo 1, Cypherpunk

Con questo tipo di remailer è possibile inviare email cifrate con PGP/GPG ed è possibile concatenare più remailer in modo che ognuno di essi conosca soltanto il remailer precedente e non tutto il percorso. Al posto del destinatario della mail va inserito l'indirizzo del server e nel messaggio va inserita l'header `Request-Remailing-To:` con l'indirizzo del destinatario.

7.2.3 Remailer di tipo2, Mixmaster

Questi remailer necessitano di un client di posta dedicato, visto che il messaggio viene costruito in maniera più complessa. Si può decidere anche la catena di remailer da utilizzare. La particolarità di questo remailer consiste nel spezzettare il messaggio in parti di identica grandezza che verrano inviate ad altri remailer e nell'inviare continuamente agli altri remailer un flusso di traffico di copertura.

I remailer anonimi erano nati soprattutto per difendere la privacy nell'invio delle e-mail ma con il passare del tempo sono stati presi di mira dagli spammer che hanno iniziato ad utilizzarli per i loro loschi scopi. Per fortuna ne esistono pochi e appena vengono scoperti vengono subito segnalati in modo da poter correggere la propria configurazione.

7.3 IP SPOOFING

L'IP spoofing è un tipo di attacco alla rete che viene utilizzato dagli hacker e che consiste nel nascondere il proprio indirizzo programmando i moduli TCP/IP del proprio software in modo tale da far apparire nella trasmissione di dati in rete un indirizzo inesistente oppure appartenente ad un'altra macchina ignara del furto di indirizzo. Per applicare questa tecnica ai loro scopi, gli spammer hanno dovuto risolvere due problemi:

- se ci si presenta al server con un indirizzo falso, non si potranno mai ricevere le sue risposte;
- trattandosi di una sessione TCP ininterrotta, il computer con indirizzo falso non potrà ricevere gli ACK, ovvero non riceverà il numero di sequenza del prossimo segmento e quindi non sarà in grado di ricostruire il messaggio.

Alcuni spammer quindi si sono avvalsi di hardware e sofware speciale per poter utilizzare lo spoofing. Più dettagliatamente basta utilizzare un computer con due interfacce di rete distinte (ad esempio una scheda di rete e un modem) in modo da poter usufruire di due indirizzi IP diversi sulla stessa macchina magari forniti anche da provider diversi. In pratica con il primo indirizzo ci si connette al server per spedire la propria posta spazzatura prestando attenzione nel cambiare in ogni pacchetto l'indirizzo del mittente con il secondo indirizzo. Le risposte del server giungeranno quindi a questo indirizzo e in questo modo anche alla macchina dello spammer che sarà in grado di ottenere i numeri di sequenza e tutte le altre informazioni necessarie per procedere tramite il primo indirizzo. Il vantaggio di questo procedimento risiede nel fatto che il primo indirizzo rimane nascosto e quindi se anche il secondo venisse scoperto e chiuso, lo spammer conserverebbe sempre l'indirizzo primario e potrebbe comunque rimpiazzare il secondo.

Per bloccare l'IP spoofing usato ai fini di spam l'unica risorsa risulta essere la prevenzione. In questo senso la RFC 2827 del maggio 2000 intitolata "Network Ingress Filtering" sancisce che i provider devono applicare un sistema di filtraggio a livello IP sui pacchetti in modo tale da bloccare quelli che presentano un indirizzo mittente che non risulta essere quello assegnato all'utente. Viene anche menzionato un filtraggio automatico sui remote access server delle reti dial-up.

7.4 PROXY INSICURI

Il proxy è un dispositivo di rete messo a disposizione da quasi tutti i provider che, tra gli altri vantaggi, offre un servizio ai clienti che consiste nel poter navigare sul web senza presentare ai siti visitati il proprio vero indirizzo IP.

In pratica il proxy viene interposto tra il client e il server, così le richieste del browser dell'utente prima di passare al server web vengono convogliate al proxy e viceversa per le risposte. In questo modo il sito visitato vede solo l'indirizzo del proxy e non quello dell'utente. Resta comunque la necessità di autenticarsi da parte degli utenti che intendano utilizzarlo. Esistono però alcuni proxy detti aperti o *insicuri* che non effettuano controlli oppure li eseguono non in modo approfondito e adeguato come dovrebbero. Questi sono spesso bersaglio di abusi soprattutto dagli spammer. Nonostante tutto si può risalire al mittente in quanto i proxy producono dei log, anche se ciò risulta sicuramente più difficile.

Sono presenti vari tipy di proxy, quelli in genere usati dagli spammer sono i proxy SOCKS che per default ascoltano sulla porta TCP 1080 e i proxy HTTP che ascoltano su porte TCP come 3128, 80 o 8080. Spesso su questi proxy gli spammer usano il comando CONNECT che permette di specificare su quale porta aprire una sessione (ovvero la 25 per poter spedire mail).

Uno strumento usato dagli antispammer per combattere il problema dei proxy insicuri è l'utilizzo di macchine *honeypot* su cui girano programmi che fanno apparire la macchina come un proxy insicuro. In questo modo quando uno spammer effettua i test per verificare se si tratta di un proxy insicuro l'honeypot risponde positivamente. Lo spammer caduto nella trappola cercherà di inviare valanghe di spam che verranno perse dato che l'honeypot non le inoltrerà. Inoltre il gestore potrà segnalare l'indirizzo dello spammer al provider.

7.5 RELAY APERTI

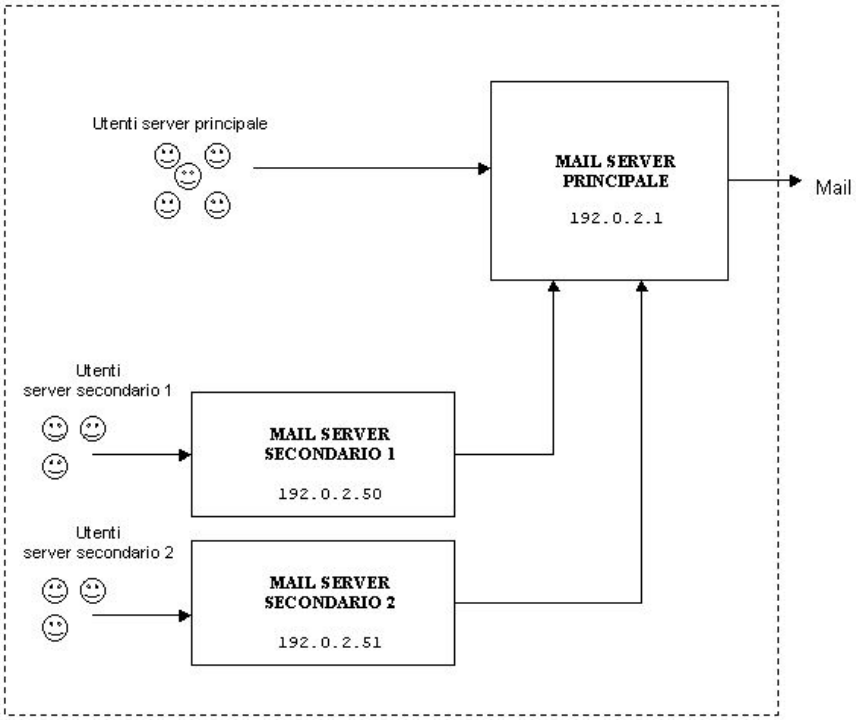
Un trucco che sta prendendo piede ultimamente nel mondo degli spammer è quello di appoggiarsi non al server del proprio provider ma a quello di un altro soggetto. Tutto ciò per evitare di essere facilmente individuabili e quindi nascondersi meglio a spese di altri server che sopportano il grosso del lavoro.

Questa tecnica è chiamata *3rd party relay*, infatti il server di un terzo soggetto ignaro viene utilizzato come relayer, ovvero come propagatore di un messaggio SMTP. E' un problema molto simile a quello dei proxy insicuri in quanto riguarda sempre il mancato controllo dell'autenticazione dell'utente. Nonostante questa funzione di relay terza parte possa essere disabilitata sui server SMTP, molti amministratori di sistema non ne conoscono l'esistenza oppure non se ne

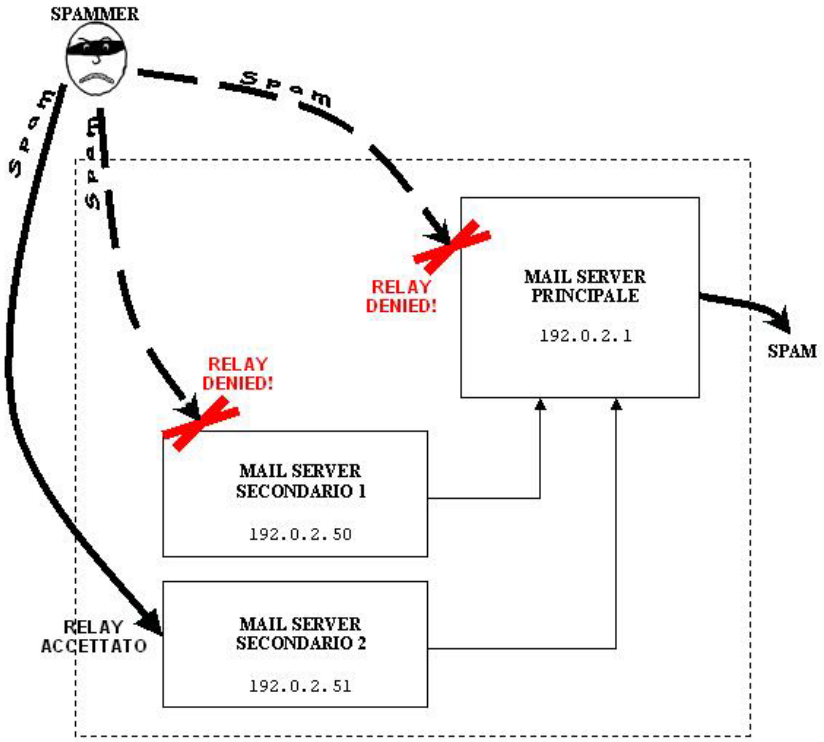
preoccupano. Allo spammer non risulta difficile individuare i relay aperti: basta dotarsi di un particolare programma automatico, adatto allo scopo, che si occupa di cercarli in rete.
La prima cosa da fare quando ci si accorge dell'abuso da parte degli spammer è disconnettere il relay aperto dalla rete per poi configurarlo in maniera adeguata. Nel caso il software non dia la possibilità di risolvere il problema è necessario sostituirlo. Ancora più efficace risulta essere la segnalazione per l'iscrizione ad una lista nera, utilizzate per bloccare la posta in arrivo da quel relay aperto.
Alcuni fra i più agguerriti antispammer si adoperano per verificare lo stato di apertura di un relay. Per farlo si basano sul fatto che i server scrivono su alcuni log le transazioni che operano incluso l'indirizzo IP da cui vengono eseguite. In pratica si tratta di inviare un'email a sé stesso utilizzando il server da verificare, ad esempio attraverso una sessione telnet. Se il server accetta il messaggio per la consegna ed effettivamente lo spedisce senza chiedere l'autorizzazione del server allora ha fallito il test ed è da considerarsi un relay aperto.

7.5.1 Relay multihop

Questo caso si presenta quando lo spam attraversa non uno ma due server SMTP. Se almeno uno dei due risulta essere aperto si parla di relay multihop. Non importa che uno dei due abbia superato i test e quindi non debba considerarsi aperto, in ogni caso ci dovrà essere un errore di configurazione che permette il relay multihop. Vediamo adesso un esempio di rete con più mail server, di cui uno è lo smarthost:



In questa rete esistono più server di cui uno principale, con un proprio bacino d'utenza ciascuno. In pratica il principale, detto anche smarthost, recapita all'esterno le email e quelli secondari per compiere la stessa operazione devono per forza passare attraverso il server principale. Basta che sono uno di questi server sia malconfigurato per rendere tutta la rete abusabile dallo spammer. Se ad esempio lo spammer prova ad inviare spam tramite il server principale e il primo secondario senza successo e trova però aperto il secondo server secondario allora i destinatari vedranno comunque provenire lo spam dall'indirizzo IP del server principale.
Vediamo quindi questo esempio di server secondario insicuro, abusabile dagli spammer:



E' necessario quindi far presente la situazione a tutti i server compresi quelli non aperti in quanto è loro interesse provvedere a correggere la configurazione del server aperto.
Sono stati riscontrati alcuni casi in cui grazie a connessioni dial-up che fungevano da relay multihop gli spammer potevano usufruire di server molto potenti e dotati di una enorme capacità di banda. In pratica gli spammer come utenti dial-up fanno girare un mail server sulla propria connessione modem. Le soluzioni a questo tipo di problema sono poche e spesso ricadono solo in una limitazione. Innanzitutto si dovrebbe limitare ad un massimo il numero di destinatari possibili per i messaggi generati da utenti dial-up e controllare periodicamente i server centrali. Ciò che risulta essere più utile è però bloccare sessioni SMTP generate dall'esterno e dirette verso nodi dial-up e

anche quelle verso nodi ad indirizzo fisso nella configurazione dei router di frontiera lasciando comunque la possibilità di sbloccarle quando un server si dimostra sicuro.
Data la vastità del problema ormai le segnalazioni a chi può sistemare il server per correggere gli errori di configurazione non hanno più molto senso, rimane solo da verificare se l'indirizzo del relay o proxy aperto appartiene ad una lista di blocco. La soluzione più estrema è quindi quella di cercare di colpire lo spammer sulle risorse a lui più care.

7.6 DIRECT-TO-MX

Gli spammer si possono dotare anche di software che simula un server SMTP per l'invio massiccio di spam. Viene definito anche "rapid fire" o RFMS (rapid fire mail server program) o anche "direct-to-MX" per il fatto che agisce collegandosi direttamente con il mail exchanger del destinatario. Il mail exchanger è un server che ha il compito di ricevere la posta in arrivo. In particolare in un record DNS (denominato record MX) del dominio viene indicato il server a cui è necessario connettersi per poter inviare posta agli utenti di quel dominio.
Viene utilizzato dai server che devono consegnare email per ricavare il nome della macchina a cui consegnarle, disinteressandosi di altri giri interni che fossero necessari per portare il messaggio fino al POP server su cui si trova la mailbox destinataria.
Un duro colpo a questa tecnica è stato dato dall'introduzione di liste dei blocchi di indirizzi IP adibiti a dial-up e dall'utilizzo del "port 25 filtering" da parte dei provider che permette di bloccare a livello dei router di frontiera qualsiasi tentativo di connessione dai nodi dial-up alla porta 25 di qualsiasi indirizzo di altre reti. Quest'ultima però si è rivelata una tecnica molto drastica e quindi non ha avuto molto successo perchè blocca anche il traffico regolare.

7.7 URL CAMUFFATE

Esistono vari trucchetti con cui vengono camuffate le URL. Innanzitutto l'indirizzo può essere espresso in notazione decimale al posto della classica quadrupla puntata a 32 bit. Ad esempio http://3221226063/ corrisponde a http://192.0.2.79/ e possono essere usati indifferentemente entrambi. L'indirizzo in base 10 è ottenuto così: 79 + (256*2) + (256^2 * 0) + (256^3 * 192). Vediamo altri esempi: http://188.1.28.79.com/ non è altro che un sottodominio del dominio 79.com oppure http://0274.01.034.0117/ è sempre http://188.1.28.79/ però questa volta è espresso in forma ottale. Altri espedienti leggermente più complicati consistono nell'introdurre una chiocciola (http://356276@202818865/abcd/def.htm dove l'host è indicato dalla parte che segue la chiocciola e corrisponde all'indirizzo 12.22.197.49) oppure più indirizzi separati dalla chioccola di cui solo

l'ultimo è quello che conta (http://209.21.162.23@23.178.75.110@216.35.27.167/eccetera/). Un ulteriore trucco consiste nell'inserire degli encoding.

Per esempio l'URL http://356276@202818865/abcd/def.htm potrebbe diventare http://3%3562%376@20%3281%3886%35/%61%62c%64/def.htm. Per ottenere l'URL giusta basta considerare il carattere % unito con i due numeri seguenti (es. %35 è il numero 5).

Alcuni link contengono anche sequenze tipo &#nn dove "nn" rappresenta il codice ASCII che esprime un carattere speciale. In pratica i caratteri speciali sono espressi con normali escape html che un browser è in grado di tradurre senza problemi automaticamente, mentre per un utente è un po' più laborioso.

Procedere manualmente a decodifare è un po' lungo e noioso, esistono siti on-line adatti allo scopo che semplificano il lavoro, come http://www.samspade.org/ .

8. COME RECLAMARE E A CHI

Purtroppo non esistono molti provvedimenti contro gli spammer. Una controffensiva sicuramente utile è la denuncia presso le autorità competenti o i provider che forniscono e possono togliere la connettività. Ciò si rivela efficace in quanto ha come obiettivo togliere allo spammer le risorse a lui più care che gli permettono di utilizzare la rete come il proprio abbonamento. E' anche vero che lo spammer potrebbe rifugiarsi altrove aprendo un nuovo abbonamento, ma con questa pratica dovrà stare fermo per un po' di tempo e non è escluso che gli venga chiuso prima o poi anche il futuro abbonamento.

Si procede innanzitutto individuando il provider dello spammer e successivamente si cerca l'esatto indirizzo del provider a cui scrivere per reclamare. L'unico rischio in cui si può incorrere nell'inviare questo tipo di segnalazione è quello di ricevere più spam e questo avviene quando lo spammer è il provider di sé stesso e quindi è lui che riceve il reclamo, ma la possibilità è molto remota. Vale la pena comunque di attivarsi per reclamare in modo da contrastare la dilagazione di questo fenomeno.

Spesso i provider nei loro contratti di abbonamento includono alcune norme sottoforma di termini e condizioni che in caso di violazione prevedono che l'utente perda il diritto ad usare il servizio, in particolar modo qualora la violazione si ripetesse dopo il primo ammonimento. A volte vengono rese disponibili alcune persone adatte proprio a leggere i reclami e a provvedere. Ci sono però anche dei provider detti spam-friendly che non si preoccupano di questo problema o addirittura aiutano gli spammer e vengono quindi bandite dalla rete attraverso le liste nere. Esistono addirittura casi in cui i provider sottoscrivono dei contratti con gli spammer, detti pink contract, che permettono lo spamming e assicurano di non prendere provvedimenti in merito, anzi si impegnano a trasmettere i reclami allo spammer. In questo modo il provider aiuta lo spammer nel listwashing ovvero nel ripulire la lista delle vittime dagli utenti che effettuano reclami e segnalazioni.

Dopo aver individuato il provider dello spammer, è necessario determinare l'indirizzo del provider a cui si può reclamare. Secondo la RFC 2142 dovrebbe seguire lo standard abuse@nomedominio ma spesso non viene seguito, e quindi bisognerà ricercare nelle pagine web del provider oppure si può consultare la lista di indirizzi di segnalazione di abusi fornita dalla Abuse.net al sito http://www.abuse.net/. Se non si dovesse trovare si può sempre scrivere a postmaster@nomedominio.it che dovrebbe essere sempre attivo.

Vediamo ora alcuni consigli sul come e cosa scrivere nel reclamo:

- Essere brevi ed essenziali, indicare chiaramente che il messaggio non era sollecitato e il motivo per cui si è giunti a loro per il reclamo;

- Nel campo “Oggetto” indicare che si tratta di una segnalazione di spam per esempio scrivendo “Posta non sollecitata proveniente da nomedominio” oppure mettere l’oggetto dell’email di spam preceduto da “Fwd”;
- Non protestare in toni accesi, è molto probabile che loro non ne abbiano colpa;
- Mettere in evidenza le aggravanti riscontrate come aver falsificato degli header, aver usato il server di una terza parte come relayer, essere un Make Money Fast (schema piramidale o schema di Ponzi) ovvero una truffa, aver già ricevuto da spam da quell’indirizzo oppure la simulazione della provenienza da un dominio esistente nel campo “`From`”che non ha nulla a che fare;
- Evitare di dire cosa devono fare, limitarsi a chiedere che vengano presi provvedimenti;
- Allegare il testo dello spam completo degli header;
- Inviare la segnalazione il più presto possibile per cercare di arginare il problema sul nascere.

Una volta mandata la segnalazione a volte si riceve una risposta automatica che conferma l’avvenuta ricezione con un numero di riferimento. Dopo la verifica nei file di log del provider che registra le attività di tutti gli utenti, verrano presi dei provvedimenti, che nel caso di prima segnalazione in genere consistono in un ammonimento. Alcuni provider poi inviano una comunicazione finale all’autore della segnalazione per confermare che si è provveduto.

Le segnalazioni da spedire in caso si riceva spam possono diventare parecchie in alcuni casi estremi. Infatti potrebbero essere coinvolti i seguenti soggetti:

1. la rete e quindi il provider da cui lo spam è stato originato;
2. la rete a cui appartiene il relay o proxy aperto da cui lo spam è transitato;
3. il servizio di hosting per il sito click-through indicato nello spam;
4. il servizio di hosting per il sito vero e proprio raggiunto tramite il click-through, se c'è;
5. il servizio di DNS per il sito vero e proprio, se diverso dal provider di hosting;
6. il gestore dei form di mailing trovati sul click-through o su altri siti;
7. il provider delle mailbox trovate nei form di mailing o come contatti sulle pagine del click-through;
8. il provider delle eventuali mailbox indicate nello spam (es. per raccogliere i remove).
9. i fornitori di eventuali servizi di altro genere (es. i servizi per addebito su carte di credito che risultano usati dai siti pubblicizzati nello spam).

Di norma comunque le segnalazioni da inviare sono poche. Le risorse più care, che per tutti gli spammer sono più difficili da rimpiazzare e su cui bisogna puntare nei reclami, sono sicuramente il sito vero e proprio e i server DNS. E’ importante comunque segnalare i casi di spam anche sulle

risorse adibite allo scopo per la pubblica archiviazione come il newsgroup news.admin.net-abuse.sightings per gli spammer internazionali e la lista Abuse del NIC per gli spammer italiani. Queste rappresentano anche utili mezzi per avere dei consulti e per verificare se per quel particolare tipo di spam è già stato preso qualche provvedimento.

Spammer con sito web

Quando in uno spam viene pubblicizzato un sito web è necessario informare dell'accaduto anche l'organizzazione che ospita o che offre connettività al sito per ottenerne la chiusura o l'ammonizione a pubblicizzarlo come spam. Il servizio DNS consente agli spammer di avere per i propri siti dei nomi che possono tenere quando vengono cacciati da un provider. Quando il sito dello spammer viene chiuso è stato reso 404 complaint ovvero quando si tenta di accederci viene visualizzato il messaggio di errore 404 Requested page not found. Sarà sicuramente più difficile per lo spammer trovare un altro servizio di hosting per riaprire il sito piuttosto che ottenere un nuovo account.

8.1 COSA DOVREBBE FARE UN PROVIDER

Prendersi carico delle segnalazioni di reclamo, attuare i vari accertamenti del caso e prendere i necessari provvedimenti va tutto a vantaggio dei provider che in questo modo acquistano successo e riescono a mantenere la propria clientela. La collaborazione per sconfiggere questa piaga dà benefici sia agli utenti sia ai provider che non saranno costretti ad aumentare i costi e quindi a perdere clientela. Questo è il motivo per cui tutti i provider si dovrebbero attivare per:

- Dotarsi di un proprio regolamento di servizio da sottoscrivere nel contratto di abbonamento o reso pubblico che vieta qualunque forma di spam pena la chiusura dell'abbonamento;
- Consentire l'accesso al servizio esclusivamente previa autenticazione;
- Aprire una casella abuse@nomedominio, comunicare poi l'indirizzo alla Abuse Net per l'inserimento nella loro lista pubblica, inserirlo tra i contatti e nel regolamento di servizio;
- Configurare con molta attenzione il proprio software per evitare soprattutto relay aperti;
- Incaricare alcune persone che si occupino delle segnalazioni ricevute;
- Adottare misure di protezione verso le mailbox dei propri utenti come black list e white list e fornire ai propri utenti filtri personalizzabili attivi sul server della posta in arrivo;
- Fornire la possibilità ai propri utenti di disporre di alias per la propria mailbox per usi diversi in modo da poterli cambiare qualora iniziassero a ricevere spam;
- Fornire informazioni e consigli sullo spam.

9. METODI PER LIMITARE LO SPAM

Esistono varie difese contro lo spam, ovvero servizi e software che vengono utilizzati dai server e dagli utenti per limitare il traffico di email indesiderate nella propria mailbox. Si possono raggruppare sostanzialmente in due tecniche distinte: il *bloccaggio* e il *filtraggio*.

Nel primo caso vengono rifiutate tutte le email provenienti da server etichettati come fonte di spam. Per quanto riguarda il filtraggio viene analizzato in modo automatico il contenuto dei messaggi e in base a particolari regole viene eliminato ciò che risulta essere spam. Entrambe hanno l'effetto di ridurre l'ammontare della posta indesiderata ricevuta mentre solo il bloccaggio permette di ridurre la banda sprecata in quanto vengono rifiutati i messaggi prima che vangano trasmessi al server dell'utente. Il filtraggio spesso risulta essere una tecnica più accurata poiché analizza tutto il messaggio nei suoi dettagli. In particolare la tecnica dei filtri si sta affinando con l'introduzione dell'apprendimento che ne aumenta l'efficacia. A volte però i filtri vengono anche etichettati come invadenti in quanto analizzando i messaggi ledono la privacy degli utenti.

9.1 LISTE DI BLOCCO (WHITE/BLACK LIST)

Le liste di blocco sono elenchi di indirizzi IP (quasi raramente di domini), selezionati secondo vari criteri. Vengono aggiornate periodicamente: vengono aggiunti indirizzi o blocchi di indirizzi e tolti quelli che non rientrano più nei parametri. Inoltre sono messe a disposizione di chiunque desideri consultarle. In genere gli indirizzi sono fonte o solo di posta legittima o solo di posta spazzatura quindi conviene tagliare fuori dalla rete solo le sorgenti di spam. Esistono anche fonti miste ma sono la minoranza. In poche parole, queste liste di blocco, chiamate anche black list, rappresentano elenchi di indirizzi IP gestiti e usati in modo da violare alcuni standard di comportamento o di sicurezza.

Questo importante strumento per limitare lo spam si rivela particolarmente utile in quanto viene applicato soprattutto a livello di server di posta e ciò aiuta a non ricevere una buona porzione di spam. La modalità principale di consultazione è l'esecuzione di una query DNS, infatti le liste sono costituite da zone DNS. In pratica si esegue il test su un indirizzo IP e si verifica se è presente o no nella lista.

Anche gli utenti possono costruire proprie liste o affidarsi a quelle già esistente. Nella scelta comunque è necessario che la lista soddisfi a due requisiti:

- ✓ sia conosciuto il criterio con cui vengono decisi eventuali inserimenti e cancellazioni dalla lista;
- ✓ sia possibile scoprire il motivo per cui un indirizzo è stato inserito attraverso una giustificazione oppure una documentazione adeguata.

9.1.1 Tipi di liste di blocco

Le liste di blocco sono state raggruppate in cinque categorie in base ai parametri che vengono utilizzati per costruirle. Vediamole in dettaglio:

1. *Liste di indirizzi o blocchi lasciati a disposizione degli spammer*

   Riguardano mail server o web server che appartengono a provider o fornitori di hosting che non si preoccupano di intervenire qualora gli venga segnalato un caso di spam che direttamente proviene da un utente di quel determinato server oppure sono stati pubblicizzati siti presenti su quel server. Inoltre fanno parte di questo gruppo anche i name server che svolgono servizio DNS per siti di spammer e organizzazioni che deliberatamente si occupano di inviare spam oppure vendono software e qualsiasi risorsa utile agli spammer. Riassumendo questa categoria è composta da chi supporta apertamente gli spammer oppure ne ignora la presenza e non si attiva per contrastare il problema

2. *Liste di relay aperti*

   Qui vengono raccolti gli indirizzi appartenenti ai relay aperti attraverso una attenta ricerca completa di test che certificano questo tipo di risorse che risultano mal configurate per ragioni tecniche, sviste o negligenza. Non sempre vengono ricercati in questo modo, più frequentemente vengono inseriti quelli di cui si ha prova che sono stati usati da spammer.

3. *Liste di risorse varie abusabili per via di buchi di sicurezza*

   In questo gruppo si parla soprattutto di proxy aperti ma anche di form-mail insicuri, ovvero script usati per inviare mail da form su pagine web. Questi ultimi vengono sfruttati dagli spammer in quanto hanno grossi limiti di sicurezza e spesso permettono di inviare enormi quantità di mail rendendo difficile però il reperimento dell'indirizzo dello spammer.

4. *Liste di blocchi adibiti a dial-up*

   In questa categoria rientrano gli indirizzi attribuiti agli utenti dinamicamente e quindi i blocchi di dial-up che vengono utilizzati dagli spammer per avvalersi di software direct-to-MX. Queste liste garantiscono la totale assenza di falsi positivi. E' buona norma considerare accettabile solo il traffico proveniente da sistemi aventi un indirizzo fisso e mutabile nel tempo.

5. *Liste con criteri particolari*

   Queste liste comprendono indirizzi che vi rientrano per vari motivi come il fatto che per un certo dominio non risulti funzionante l'indirizzo di posta "postmaster" o "abuse" oppure il fatto che i dati presenti nel database WHOIS siano riconosciuti come falsi oppure mail server che hanno una configurazione non conforme alle RFC.

9.1.2 Alcune liste di blocco disponibili

Inizialmente le liste di blocco venivano costruite da amministratori di rete esclusivamente ad uso privato. Tra le prime rese pubbliche sicuramente RBL (Realtime Blackhole List) gestita dal MAPS (Mail Abuse Protection System, http://www.mail-abuse.com/) è una delle più importanti e diffuse. RBL è una lista di indirizzi o blocchi di indirizzi che risultano essere a disposizione degli spammer. Può essere distribuita sia come zona DNS che sottoforma di BGP, ovvero un protocollo di routing utilizzato per far comunicare i router in modo da determinare gli instradamenti dei pacchetti tra varie reti. L'utilizzo tramite BGP avviene solo nelle reti che hanno deciso di servirsene e provoca limitazioni anche per il traffico non e-mail.

L'apporto derivato da MAPS nella costruzione di liste di blocco per la lotta antispam si è rivelato particolarmente efficace anche per il tentativo di educare le reti trasgredenti, attraverso un contatto diretto che se non va a buon fine porta all'iscrizione dell'indirizzo nella lista. Fornisce inoltre utili informazioni per risolvere il problema dei relay aperti e una lista di relay aperti (RSS), che contiene solo quelli che sono stati segnalati come fonte di spam, dopo opportuni accertamenti. Non si preoccupa di effettuare test per eventualmente scoprire queste mal configurazioni, a differenza di ORBS, che è una lista che si occupa proprio di questo, la quale però fu chiusa nel maggio 2001 a causa di problemi della persona che la gestiva. MAPS ha introdotto anche una lista di dial-up (DUL). Dopo una serie di problemi anche giudiziari, a partire dall'agosto 2001, le liste diventarono accessibili solo per i sottoscrittori di un contratto con MAPS. Questa soluzione declinò il successo di MAPS.

Vediamo ora un elenco di possibili liste a cui far riferimento, con la dovuta cautela e con l'obbligo di mantenersi aggiornati sulla situazione. Sono suddivise in categorie, anche se alcune di esse possono far parte di più classificazioni.

Liste di relay e/o proxy aperti

E' necessario fare molta attenzione e non scegliere liste che bloccano smarthost (mail server principali), si correrebbe il rischio di incorrere in falsi positivi. Sicuramente le più importanti in questo campo sono:

- ORDB, http://www.ordb.org/
  Lista di relay aperti single-hop, che dà la possibilità di leggere i messaggi di test che evidenziano lo stato di apertura.
- Distributed Sender Boycott List, http://dsbl.org/
  Lista che riguarda sia relay che proxy aperti e fornisce una documentazione dei test.

- Not Just Another Bogus List, http://www.njabl.org/

  Lista che comprende non solo relay aperti e proxy insicuri ma anche sorgenti di spam, indirizzi IP assegnati dinamicamente e relay multihop, sempre con disponibilità di leggere i messaggi di test.
- Composite Blocking List, http://cbl.abuseat.org/

  Vengono listati indirizzi IP da cui è provenuto spam e proxy insicuri, è mantenuta tramite spamtrap (indirizzi e-mail usati come trappole per lo spam). Non fornisce documentazione ed è distribuita anche da Spamhaus.org con il nome XBL.

Liste di blocchi adibiti a dial-up

Comprende liste di indirizzi IP che vengono allocati dinamicamente. La più importante è sicuramente DUHL (http://www.dnsbl.au.sorbs.net/), fornita da Sorbs.net.

Liste di indirizzi o blocchi lasciati a disposizione degli spammer

- SBL (Spamhaus Block List), http://www.spamhaus.org/

E' una lista creata e gestita in Inghilterra da un antispammer molto esperto, Steve Linford. Si basa sulla convinzione che lo spam è in mano a grosse centrali di spammer. Fornisce informazioni sui netblock utilizzati, sui backbone che li connettono e così via. L'efficacia di SBL si è rivelata sempre più consistente grazie anche alla focalizzazione su spamhaus di grandi dimensioni e all'attenzione rivolta nel non determinare falsi positivi.

- SPEWS (Spam Prevention Early Blocking System), http://spews.org/

I gestori di questa lista si mantengono nell'anonimato. Si prefigge il compito di risolvere i problemi riscontrati in RBL e si adopera per individuare gli spammer appena iniziano la loro attività e a volte anche prima. La lista è disponibile ed utilizzabile tramite query DNS oppure download dei files di zona. In particolare è formata da due liste: level1, che è quella di blocco, e level2 che contiene indirizzi sotto osservazione. Ha una politica molto aggressiva, infatti se lo spam continua si riserva di aumentare il blocco di indirizzi iscritto nella lista, con il rischio di comprendere anche utenti che non hanno nulla a che fare con lo spam. Tuttavia si rivela molto efficace soprattutto nei casi in cui educare è inutile.

Liste di indirizzi su base geografica

- The South Korea Blocking List, http://korea.services.net/

  Si è deciso di introdurre questo tipo di lista in quanto dalla Corea proviene una quantità decisamente cospicua di spam, soprattutto perché abbondano proxy insicuri. Comprende sia

spam locale che americano o di altra provenienza. Altri paesi in cui si è adottato questo tipo di lista sono la Cina, la Nigeria e il Brasile. Si tratta di una soluzione molto veloce con buoni risultati.

9.2 FILTRI ANTISPAM

I filtri antispam si basano su regole che si applicano al testo delle e-mail ricevute per stabilire se si tratta di spam. Andremo ad analizzare il funzionamento e i vari tipi di filtraggio ma prima di tutto è necessario distinguere due tipi di filtri:

- Filtri sul client: in questo caso sono applicati sul client di posta e quindi gli header dei messaggi vengono scaricati tutti, poi verrà controllato il contenuto e quindi scaricato solo quello che non è spam. Questo comporta perdita di tempo e soprattutto non fa risparmiare al server il costo del messaggio.
- Filtri sul server: in questo caso sono applicati sul server e ciò risulta più efficace in quanto è il server che si occupa della verifica, infatti all'utente vengono presentati solo i messaggi legittimi che hanno superato il controllo. Spesso però stabiliscono solo se si tratta di spam senza eliminare il messaggio.

In relazione a questo si possono quindi avere:

- Mailbox prive di qualsiasi filtro: purtroppo riguardano la maggior parte dei provider.
- Mailbox con possibilità di settare filtri sul mittente (campo "From"): Spesso sono abbinate alle whitelist, ovvero le liste contenenti gli indirizzi delle persone da cui si vuole ricevere posta bloccando tutto il resto del traffico in entrata che non provenga da quegli indirizzi.
- Mailbox con possibilità di settare filtri su vari campi.
- Mailbox con possibilità di settare filtri sul server di provenienza: in questo caso vengono adottate le blacklist che bloccano il traffico proveniente da server che risultano fonte di posta indesiderata.

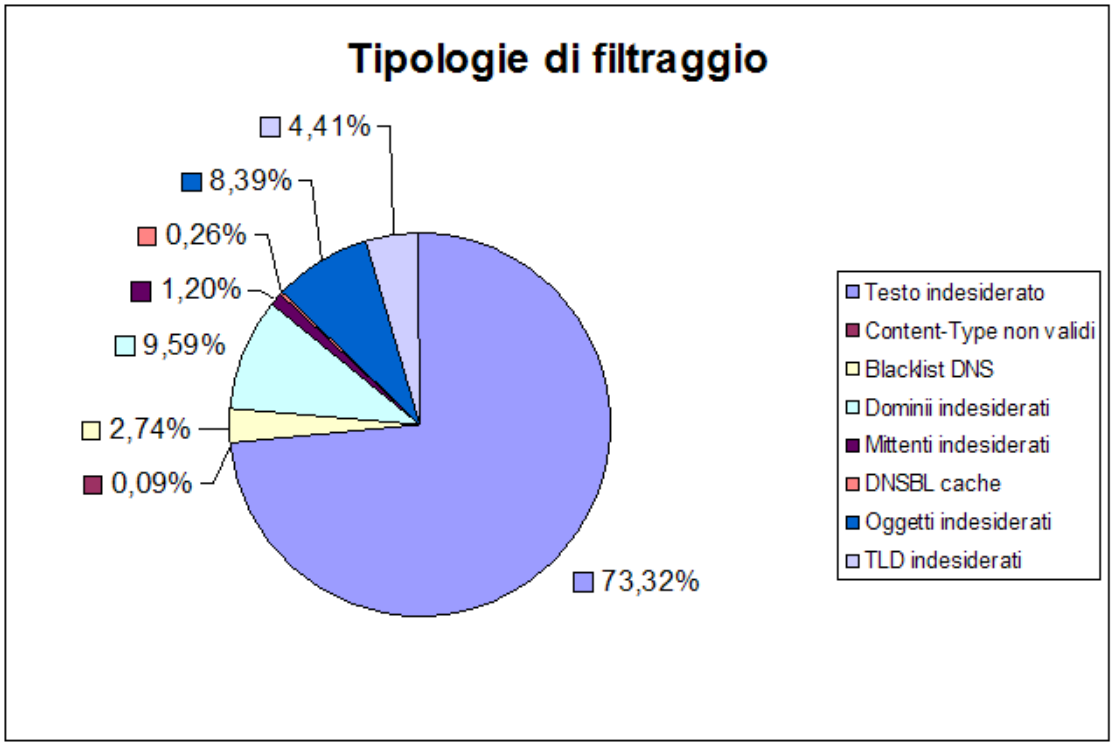
Svantaggi e vantaggi rispetto alle liste di blocco

Innanzitutto è un metodo che si può applicare solo dopo aver ricevuto lo spam e quindi solo dopo aver individuato il tipo di filtro più adeguato e sopratutto viene scaricato sull'utente l'onere di occuparsi del blocco delle e-mail illegittime. Seconda cosa per utenti inesperti risulta essere un sistema un po' difficile da applicare e inoltre richiede tempo per l'aggiornamento e per il raffinamento dei filtri.

Il problema principale dei filtri è l'affidabilità e quindi il rischio di creare falsi positivi, ovvero bloccare erroneamente e-mail legittime che ricadono nelle regole impostate senza contare che non

sempre si riesce ad individuare tutti i messaggi di spam soprattutto perché gli spammer affinano sempre di più le loro tecniche. Rispetto alle liste di blocco limita di più la connettività in rete dell'utente, invece di bloccare quella degli spammer.

Nonostante ciò il loro vero pregio consiste nel bloccare in modo molto più consistente la ricezione di spam rispetto alle liste di blocco come si può vedere nel grafico che riassume le tipologie di filtraggio in ordine di efficacia. In questo grafico si tiene comunque conto del fatto che un messaggio spam può ovviamente soddisfare una o più delle condizioni (ossia può contenere parole "vietate" nel testo, ed anche nell'oggetto), ognuna delle quali determina un "hit" ossia un conteggio su tale tipologia.



Notiamo che la stragrande maggioranza del filtraggio viene effettuato in base al corpo del messaggio. Questo fatto è diretta conseguenza dei vari tentativi di "mascheramento" del contenuto da parte degli spammer, i quali tendono a non inserire più parole "pericolose" nell'oggetto o come mittente. Il secondo metodo (con quasi il 10%) è per "dominio indesiderato" ossia il mittente o uno dei server per i quali il messaggio e' transitato appartiene ad uno dei nomi inseriti in lista. Il "vecchio" metodo di filtraggio per oggetto risulta solo il terzo in ordine di importanza. I Top Level Domain (TLD) raggiungono il 4,41%, seguono poi le Blacklist DNS, i mittenti indesiderati e le DNSBL. Infine con una piccolissima percentuale (0.9%) contribuiscono anche i Content-Type non validi.

9.2.1 Tipi di filtraggio

Col passare degli anni si sono sviluppate tre tecniche distinte di filtraggio:

- Statico
- Euristico
- Statistico o probabilistico

Filtraggio statico

Nel primo caso, che rappresenta anche il primo sistema di filtraggio che è nato, gli amministratori di sistema specificano liste di parole oppure espressioni regolari che spesso sono presenti nei messaggi di spam in modo che il server scarti tutte le e-mail contenenti quelle parole. Questa tecnica però ha il suo rovescio della medaglia: infatti risulta particolarmente difficile l'aggiornamento e soprattutto la tendenza a procurare falsi positivi (ovvero e-mail regolari erroneamente scambiate per spam) purtroppo è abbastanza consistente.

Filtraggio euristico

Il filtraggio euristico si basa sull'assegnare un punteggio numerico a frasi o modelli che si presentano nel messaggio. Se è positivo indica che probabilmente il messaggio contiene spam, altrimenti se è negativo si tratta di un messaggio di posta legittimo. Viene impostata inoltre una soglia del punteggio in modo da poter stabilire se i messaggi vengono rifiutati o segnalati come spam se superano questo limite. In ogni caso è sempre l'amministratore che gestisce le liste di punteggi associate ai messaggi e i criteri con cui assegnare i punteggi.

Filtraggio statistico

Il filtraggio statistico è stato proposto per la prima volta nel 1998 nel AAAI-98 Workshop on Learning for Text Categorization, ed è stato reso popolare da Paul Graham nel 2002. Usa metodi probabilistici che si fondano sul Teorema di Bayes, in modo da poter predire se un messaggio è spam. Necessita un'iniziale fase di apprendimento in cui vengono sottoposte al filtro raccolte di –e-mail legittime e indesiderate ricevute dall'utente.

Da qualche tempo stanno crescendo anche vari sistemi di filtraggio che comprendono più tecniche di riconoscimento dello spam combinandole in modo da cercare di minimizzare il rischio di falsi positivi e per aumentare l'efficienza del filtraggio.

9.2.2 Funzionamento

Solitamente i filtri sono impostati in modo da riconoscere il campo del mittente o del destinatario oppure vengono implementati per individuare specifici vocaboli nell'oggetto o nel corpo del messaggio di posta, in modo da poter scegliere l'azione da compiere. Infatti il messaggio può essere eliminato o memorizzato in una cartella particolare dedicata allo scopo messa a disposizione dell'utente che deciderà come occuparsene.

Il procedimento consiste nell'attribuzione di un punteggio alle e-mail in base ad alcuni parametri che riguardano soprattutto le informazioni che contiene. Un punteggio positivo determina il rischio che l'email sia indesiderata, mentre quello negativo riscontra la legittimità dell'e-mail.

Per attribuire il punteggio vengono seguiti alcuni criteri:

- Analisi dell'intestazione per determinare se l'e-mail proviene da un client di posta oppure da un sistema automatico.
- Analisi del contenuto e verifica se vengono violate alcune regole fisse: ad esempio le e-mail scritte in HTML, con caratteri grossi e molto colorate fanno lievitare il punteggio oppure la presenza di termini particolari come "remove", "price", "enlargement".
- Analisi del contenuto con regole dinamiche e statistiche: è il caso dei filtri bayesiani, che hanno bisogno di una sorta di "allenamento" attraverso la classificazione delle e-mail giuste e quelle indesiderate prima di funzionare ad ottimi livelli.
- Ricerca del pattern associato all'e-mail in database distribuiti in rete che si occupano di raccogliere informazioni su tutto lo spam che circola in rete.

Viene poi stabilita una soglia ed in base a questa se il punteggio risulta superiore si tratta di spam. In alcuni casi poi si possono creare due cartelle di posta, una contenente le e-mail legittime e l'altra lo spam. Questo avviene soprattutto nei filtri bayesiani al fine di migliorarne l'autoapprendimento e l'efficacia.

9.2.3 Un particolare tipo di filtri: i filtri bayesiani

Questo filtro, denominato Bayesiano, è di recente diffusione e sta sbaragliando la concorrenza in quanto dà risultati molto più soddisfacenti rispetto a quelli tradizionali. Questa nuova tecnica di filtraggio è nata soprattutto per tentare di arginare il problema degli altri filtri. Infatti il metodo delle parole chiave è facilmente aggirabile dagli spammer: basta storpiare la grafia delle parole (*"p0rn0"* e "*v-i-a-g-r-a"* sono esempi classici) oppure usare ogni volta indirizzi diversi per il mittente. L'eventualità che i messaggi di spam riescano a superare i controlli non è quindi remota e anche il rischio di cestinare messaggi utili senza cancellare quelli indesiderati è altissimo.

I filtri bayesiani si fondano sul Teorema di Bayes, scritto circa trecento anni fa dal matematico Thomas Bayes. Questo teorema è stato adattato al contesto dello spam ed in particolare enuncia che la probabilità che un'e-mail sia spam è data dal contenuto di essa, ovvero da specifiche parole presenti, ed è uguale alla probabilità di trovare queste parole nell'e-mail moltiplicata per la probabilità che ogni e-mail sia spam e diviso poi il tutto per la probabilità di trovare quelle parole in ogni e-mail. Vediamo la formula:

$$P(spam \mid parole) = \frac{P(parole \mid spam) * P(spam)}{P(parole)}$$

Si può anche considerare le parole del testo indipendenti, cosa non vera ma che dà buoni risultati pratici. Quindi il messaggio:

$X = [x_1, \ldots, x_n]$

viene trattato come un vettore di variabili aleatorie indipendenti ed equidistribuite; ogni variabile aleatoria ha come realizzazione una parola del vocabolario. Dall'indipendenza segue che (considerando l'evento spam):

$P[X = spam \mid X=(a_1, \ldots, a_n)] = \prod_i P[X = spam \mid a_i = x_i]$

dove le a_i sono semplicemente le parole del messaggio ricevuto. A questo punto si applica la regola di Bayes:

$$P[X = spam \mid a_i = x_i] = \frac{P[a_i = x_i \mid X = spam]}{P[a_i]} * P[X = spam]$$

Si applica la teoria frequentista e si calcolano le frequenze delle parole contenute nei messaggi riconosciuti dall'utente come spam:

$P[a_i = x_i \mid X = spam]$ è la frequenza della parola a_i in tutti i messaggi di spam.

Le frequenze delle parole si possono trovare facilmente:

$P[a_i]$ è la frequenza a priori della parola a_i.

Le frequenze dei messaggi di spam sono banalmente calcolabili:

$P[X = spam]$ è la frequenza dello spam

Conoscendo quindi questi dati si ricava la probabilità che il messaggio sia spam:

$P[X = spam \mid a_i = x_i]$.

Esistono particolari parole che hanno probabilità di essere presenti nelle e-mail di spam. Fra queste per esempio la parola "Viagra" è una delle più gettonate, mentre raramente si trova nelle e-mail regolari.

Il filtro non conosce inizialmente questi vocaboli e quindi deve effettuare prima una sorta di "allenamento". In poche parole, dopo una breve sessione di addestramento in cui si sottopone al filtro campioni di spam ricevuto, questo è in grado attraverso un'analisi statistica di indicare la probabilità che le e-mail ricevute siano posta indesiderata. Vediamo di analizzare questo procedimento più in dettaglio.

L'utente deve manualmente indicare se una nuova e-mail è spam o legittima e dividere le e-mail in due cartelle: una contenente spam e l'altra e-mail legittime (dette anche ham). Analizzando queste cartelle il filtro aggiornerà il suo database contenente le parole indesiderate e quelle regolari in modo da aggiustare le varie probabilità con cui si presentano. Viene analizzata automaticamente la frequenza d'uso delle varie parole contenute, includendo anche i codici HTML e i dettagli delle intestazioni. Più è frequente una parola nel campione di spam analizzato, più è probabile (ma non certo) che ogni messaggio che la contiene sia spam.

Dopo questa fase di apprendimento, il filtro classifica i messaggi in base alla probabilità complessiva delle varie parole che contengono. Per esempio, se un'e-mail contiene una parola ad alto rischio spam ma per il resto è costituito da parole poco usate dagli spammer, non viene classificato come spam. Ogni parola contribuisce ad incrementare la probabilità e influisce sul risultato finale. Se il totale eccede una certa soglia il filtro contrassegnerà l'e-mail come spam e a questo punto sarà automaticamente spostata in una cartella contenente e-mail indesiderate o cancellata definitivamente.

Il filtraggio bayesiano quindi si basa su un concetto relativamente semplice e cioè sull'apprendimento automatico e sul fatto che gli spammer per promuovere il loro prodotto non possono fare a meno di usare determinate parole come "visita", "clicca", "compra", "rimborsati", "promozionale", "offerta" e così via. L'apprendimento automatico fornisce la possibilità di superare il problema di generare automaticamente elenchi di parole proibite con relative varianti ortografiche inventate dagli spammer per storpiare le parole ed evitare i controlli. Semplicemente basta sottoporre al filtro un messaggio e indicargli se è o no spam. Le parole più ricorrenti verranno automaticamente considerate indicatori di spam alzando quindi il punteggio e quelle meno frequenti verranno considerate indizi scagionanti e contribuiranno ad abbassare il punteggio. In questo modo una mail legittima che contiene anche solo una parola attinente alla posta spazzatura non verrà classificata come spam grazie all'abbassamento del punteggio che garantiscono le altre parole.

Più spam gli viene dato in pasto, più il filtro diventa preciso. Visto che le parole chiave non sono predeterminate ma dinamiche, il filtro impara giorno per giorno le nuove tecniche degli spammer e tiene sempre aggiornata la protezione della macchina, ma il rovescio della medaglia è evidente: perché il filtro dia il meglio di sé è importante, anzi vitale, che l'utente collabori, e indichi al

sistema cosa è spamming e cosa no. Almeno finché il filtro non può camminare con le sue gambe. Un vantaggio da non sottovalutare consiste nel fatto che il filtraggio attraverso questo tipo di filtri offre la possibilità di allenarli sulla base di ogni utente e quindi l'analisi risulta focalizzata, specifica e personalizzabile. Si basa sul tipo di spam che si riceve e non su quello che circola in rete. In questo modo ognuno può decidere quali messaggi di posta ritenere indesiderati e quali no e ciò si rivela molto utile e rende il filtro ancora più efficace a livello utente evitando errori di valutazione.

Più in generale, come nota Paul Graham, la sorpresa di quest'analisi dello spam è che gli spammer utilizzano in realtà un sottoinsieme piccolissimo dei vocaboli di una lingua. Sono terribilmente ripetitivi (c'è un limite al numero di modi per dire "compra il mio prodotto"), ed è questo loro tallone d'Achille che consente a un sistema automatico di ottenere questi risultati eccezionali.

Recentemente gli spammer hanno sviluppato tattiche che includono nel messaggio parole random che non hanno nessun attinenza con il testo e soprattutto con lo spam per cercare di eludere questi controlli e quindi diminuire il punteggio dello spam aumentando invece quello di ham. Purtroppo ciò rende questi messaggi quasi immuni ai filtri diminuendo l'efficacia del filtro.
Tuttavia questa tecnologia ha enormi margini di ulteriore miglioramento. Si pensa già per esempio di analizzare la frequenza delle coppie di parole anziché i singoli vocaboli.

Confronto con gli altri filtri e le liste di blocco

Rispetto ai filtri che ricercano parole chiave e alle liste di blocco i filtri bayesiani offrono una migliore percentuale di precisione nel bloccare lo spam e riconoscere le e-mail legittime, la quale arriva fino addirittura al 99,5% e con un'incidenza di falsi positivi che si aggira attorno allo 0,3%. Un gran bel risultato se si considera che un programma antispam convenzionale di solito riesce a bloccare percentuali attorno al 90% con falsi positivi che oscillano attorno all'1-3 %. L'approccio dei filtri tradizionali infatti raggiunge la quota del 70%-80% e rende quasi impossibile riconoscere il resto della posta. Restringendo i parametri del filtro si rischia di ottenere falsi positivi. L'approccio a white/black list presenta lo svantaggio di dover gestire manualmente l'elenco dei mittenti qualora non ci si affidi alle liste di blocco pubbliche, le quali però bisogna accertarsi che risultino aggiornate periodicamente. Inoltre i server che ospitano le «Black list», cioè le liste di spammer conclamati, sono il bersaglio preferito dagli hacker, e ormai non catturano più del 25% dello spamming, oltretutto con una mostruosa capacità di far danni coi falsi positivi. Per contro le «White list», i meccanismi che consentono l'ingresso solo ai messaggi degli utenti certificati funzionano abbastanza bene, ma da un lato sono un formidabile ostacolo a nuovi contatti commerciali e dall'altro possono essere aggirati facilmente da messaggi che «impersonano» gli utenti registrati. Grazie alla sua larga diffusione e al suo successo nel combattere lo spam, si è deciso di applicare

questa tecnica anche per classificare ogni tipo di dato. Sono infatti comparsi i primi programmi di classificazione, basati su questa tecnica, in campo scientifico, medico e ingegneristico.

Prodotti che sfruttano i filtri bayesiani

Sono molti i prodotti che sfruttano i filtri bayesiani, e parecchi possono essere scaricati gratuitamente da Internet; una buona raccolta è pubblicata sulle pagine di Graham (http://paulgraham.com/filters.html).

In particolare, Mozilla Mail consente di usare diversi tipi di filtri plugin contro le mail indesiderate, anche se al momento uno solo è supportato: guarda caso, proprio quello che offre la classificazione bayesiana.

Come tutti i filtri di questo tipo, nei primi giorni di utilizzo bisogna dare al sistema esempi di mail buone e indesiderate, e piano piano il filtro comincia a funzionare da solo; ma anche una volta raggiunto un buon risultato è importante continuare a classificare lo spam, così per tentativi ed errori il filtraggio diventa sempre più accurato.

Anche Apple Mail ed Eudora 6 sfruttano filtri bayesiani, mentre per gli utenti di Microsoft Outlook ci sono diversi plugin di terze parti, come SpamBully (http://www.spambully.com/) che è compatibile anche con Outlook Express; di SpamBayes (http://spambayes.sourceforge.net/) è disponibile il plugin 008 per Outlook, che ha il non secondario vantaggio di essere gratis e open source, così come PopFile (http://popfile.sourceforge.net/).

SpamTunnel (http://uiorean.cluj.astral.ro/), arrivato alla versione 1.01, è anch'esso gratuito ed essendo basato su java è compatibile con Windows, Linux, Mac OS X e Solaris.

A tanta flessibilità non corrisponde però un altrettanto buona semplicità d'uso; durante la delicata fase della categorizzazione, i messaggi vanno spostati a mano nelle caselle «bad» e «good». Per facilità d'uso e prestazioni, tra i migliori filtri bayesiani gratuiti vanno segnalati dunque SpamBayes e PopFile, che affidano a una chiara interfaccia Web la fase di training, durante la quale imparano a distinguere tra spazzatura e messaggi validi.

10. TOOLS, STRUMENTI SOFTWARE

Per monitorare ed eliminare lo spam è necessario utilizzare antispam software. In rete è possibile reperire software in grado di controllare la posta indesiderata. Questi programmi sono generalmente shareware o freeware ed il download è gratuito.

I programmi antispam hanno raggiunto un livello tale da permettere di eliminare più del 90% dei messaggi di spam.

Elenchiamo ora i principali software antispam in circolazione che poi andremo ad analizzare in dettaglio:

- Spam Assassin
- Spam Terminator
- Mail Washer
- Sam Spade
- SpamPal
- Spamihilator

10.1 SPAM ASSASSIN

Spam Assassin rappresenta uno dei migliori software antispam in circolazione oggi ed è universalmente riconosciuto come una delle migliori difese contro lo spam. Si tratta di un prodotto open source direttamente scaricabile all'indirizzo http://www.spamassassin.org/ .

Il pacchetto è scritto in Perl e, fino alla versione 2.6x, è stato rilasciato sotto la Perl Artistic License. A partire dalla versione 3.0 il progetto è entrato a far parte della Apache Software Foundation e viene distribuito con la stessa licenza di Apache. Il suo compito è quello di estrarre da ogni messaggio quante più informazioni può, elaborarle, confrontarle ed emettere un verdetto. Per questo si limita a visionare una per una le mail che gli vengono sottoposte, per poi apporre il suo timbro.

SpamAssassin è un mail filter che permette di identificare lo spamming attraverso controlli multipli che comprendono l'analisi del testo e degli headers di un messaggio di posta anche attraverso espressioni regolari (regexp); oltre ad appoggiarsi a RBL pubbliche, a liste Black e White automatiche, si avvale anche di filtri bayesiani e database basati su spam-tracking collaborativo come Razor, DCC (http://www.rhyolite.com/anti-spam/dcc) e Pyzor (http://pyzor.sourceforge.net). Vediamo in dettaglio queste caratteristiche:

- Analisi dell'intestazione: gli spammers usano numerosi trucchi per mascherare la loro identità, invitandovi a credere che l'e-mail giunta sia valida, o facendovi credere di esservi iscritti voi stessi a determinate liste. SpamAssassin individua questi casi trovando le incongruenze tipiche della posta generata automaticamente.
- Analisi del testo: le e-mail spam hanno determinate caratteristiche di stile testuale e ricorrenze tipiche. SpamAssassin riesce ad individuare anche questi.
- Blacklist: SpamAssassin supporta molte blacklists aggiornate, quali ad esempio mail-abuse.org e ordb.org.
- Razor (http://razor.sourceforge.net): Razor è un database che tiene traccia dello spam, si occupa di inserire nell'archivio una copia dei messaggi di spam. Poiché lo Spam funziona tipicamente trasmettendo un messaggio identico a centinaia di persone, Razor lo blocca permettendo alla prima persona che riceve un'e-mail spam di aggiungerla al database. Gli invii successivi saranno così bloccati. In pratica ogni mail viene confrontata con quelle presenti nel database e se corrisponde Spam Assassin le assegnerà un punteggio molto alto altrimenti può inviarla a Razor per ampliare il database in modo che tutti gli altri ne beneficino.

Tutto ciò permette di aumentare la probabilità di individuare esattamente i messaggi indesiderati e diminuisce la possibilità di incorrere in falsi positivi. L'efficienza raggiunta dal software è piuttosto elevata: arriva al 99,6%. Unico neo è la piccola percentuale di falsi positivi che si ferma allo 0,1%, cifra irrisoria considerando l'innumerevole posta spazzatura che Spam Assassin ci permette di eliminare.

Il metodo di valutazione se un messaggio è da considerarsi spam è euristico, vengono assegnati diversi punteggi (configurabili) a diversi elementi e si valuta il punteggio restituito da ogni test sulle e-mail, il quale può essere positivo o negativo. Per impedire agli spammer di trovare il modo di eludere i test di SpamAssassin, il progetto include il maggior numero di test possibile ed è inoltre facilmente estendibile. Ciascun test ha un determinato "peso" determinato esaminando un campione significativo di messaggi di spam da cui sono state estratte le caratteristiche salienti; grazie all'aggiornamento continuo il programma diventa sempre più intelligente e, quindi più efficace.

I test controllano tre categorie di elementi: il contenuto dei messaggi, le caratteristiche tecniche e le informazioni di fonti esterne. Fanno parte della prima categoria le verifiche sul soggetto e sul testo che riguardano ricerca di parole o frasi tipiche. Purtroppo, per ora, l'italiano non è una lingua supportata e quindi questi primi filtri non danno risultati, anche se poi tutto si aggiusta grazie agli altri metodi. I controlli tecnici sono i più raffinati e verificano tutto ciò che può dare un indizio sul fatto che le mail siano state preparate automaticamente come ad esempio validità della data e del

fuso orario indicato, tipo di programma usato per la spedizione, colore, dimensione del testo, caratteri inconsueti e formato dell'indirizzo di provenienza. Il colpo di grazia viene dalla terza categoria di test, quella che si appoggia a siti esterni ovvero alle liste aggiornate sui server che è risaputo che consentono l'invio di spam o non seguono determinati criteri di prevenzione. Ne sono alcuni esempi i siti http://www.rfc-ignorant.org e http://www.osirusoft.com.

La somma di questi test viene confrontata con una certa soglia limite e se questa viene superata si considera il messaggio come spam. In particolare Spam Assassin aggiunge all'intestazione di ogni messaggio un campo denominato *X-Spam-Level:*, che mostra il "livello" di spam del messaggio, calcolato in base a regole interne del programma. Più asterischi contiene tale campo e più alta è la probabilità che si tratti di un messaggio contenente spam. Se il numero di asterischi presenti in questo campo é maggiore o uguale a 5, il messaggio è riconosciuto come spam e viene "marcato" aggiungendo al soggetto la stringa "*****SPAM*****". Inoltre vengono inserite altre tre intestazioni:

- *X-Spam-Flag:* impostato a YES indica che il messaggio è considerato spam;
- *X-Spam-Status:* indica con `hits=` il punteggio ottenuto, con `required=` il limite per considerare il messaggio spam, con `tests=` i tests che sono stati applicati, con `autolearn=` se è stato applicato l'autolearning per i filtri bayesiani e con version= la versione di SpamAssassin installata;
- *X-Spam-Checker-Version:* indica la versione di SpamAssassin installata.

Spam Assassin ha un'altra caratteristica molto importante: può ricordare il comportamento di un mittente e modificare il giudizio di conseguenza, dandogli credito o sfiduciandolo. Si avvale, infatti, di una whitelist automatica che per ogni individuo tiene traccia della media dei punteggi ottenuti. Ad ogni mail ricevuta si controlla anche quel valore in modo che così il punteggio medio si può spostare in positivo o negativo. Per attivare l'auto-whitelist è necessario che SpamAssassin venga attivato con l'opzione `-a`.

A seconda dello scenario, l'utente o l'amministratore sono liberi di modificare alcuni semplici parametri di configurazione per ottenere un'azione di filtraggio più decisa o conservativa a seconda delle esigenze. Ad ogni modo, spamassassin si limita a marchiare lo spam, mentre la gestione delle e-mail identificate come spam (es. cancellazione o archiviazione in cartelle apposite in modo che se ne possa valutare il contenuto per evitare di eliminare messaggi corretti che altrimenti andrebbero perduti) è demandata ad altri programmi (es. procmail) e può essere adattata alle proprie esigenze. La configurazione di base è piuttosto semplice e i parametri di default sono validi per molte occasioni. Spam Assassin è destinato principalmente all'utilizzo su mail server, campo nel quale ha

ormai una solida e meritata reputazione. Tuttavia è il caso di ricordare che in molti sistemi linux la consegna in locale della posta è comunque gestita in un qualche modo da un Mail Transfer Agent, e in tutti questi scenari Spam Assassin è facilmente integrabile e si presta pertanto anche all'uso da parte dell'utente di casa.

Per facilitare il filtraggio di grosse quantità di posta ed il collegamento al maggior numero possibile di caselle, Craig Hughes ha scritto il demone "spamd", distribuito col pacchetto di SpamAssassin.

Il più grosso punto debole di SpamAssassin è quello di essere rivolto ad utenti tecnicamente preparati e di non avere (ancora) una interfaccia grafica.

Sistemare questa cosa, così come scrivere più test e creare più moduli per i sistemi di posta sono le cose più importanti per un ulteriore sviluppo. Sono attualmente disponibili collegamenti a Procmail, Qmail, Postfix e Sendmail attraverso la libreria Milter ed il plug-in Mail::Audit.

Riassumiamo quindi i principali vantaggi e svantaggi di Spam Assassin.

Vantaggi:

- Molto efficace nell'identificare lo spam
- Ottimo per mail server
- Ampia configurabilità
- Ben documentato

Svantaggi:

- Necessita di altri programmi per gestire lo spam identificato

Approfondiremo meglio i dettagli sull'installazione e configurazione di Spam Assassin nella parte pratica che si occuperà proprio di questo, mentre tralasceremo quelli relativi agli altri strumenti software che potranno essere comunque reperiti attraverso i siti indicati.

10.2 SPAM TERMINATOR

Spam Terminator rappresenta il contributo italiano per la lotta allo spam, è gratuito ed è scaricabile all'indirizzo http://www.spamterminator.it/download.asp. Agisce ancor prima di scaricare i

messaggi e anch'esso funziona come server proxy POP3 trasparente. In pratica quando il client di posta cerca di collegarsi al server Spam Terminator intercetterà la chiamata e si attiverà per ripulire direttamente dal server i messaggi indesiderati che non superano il controllo dei suoi filtri. In questo modo la cesella di posta dell'utente sarà notevolmente alleggerita comprendendo solo mail legittime secondo i criteri di Spam Terminator.

Da notare un piccolo difetto di questo programma che consiste nell'impossibilità di recuperare i messaggi scartati mediata però dal salvataggio delle header degli stessi al fine di identificare le e-mail che sono state eliminate. La tecnica di filtraggio si basa sulle intestazioni dei messaggi. Per quanto riguarda il campo del mittente "`From`" è possibile verificare se è corretto, se è vuoto e se non appartiene alla lista dei mittenti leciti. Per il campo del destinatario "`To`" vale lo stesso discorso, ovvero si controlla che l'indirizzo presente non appartenga a liste proibite e che non sia uguale a quello del mittente. Nel campo "`Subject`" non devono comparire particolari termini, caratteri o spazi usati dagli spammer e questo vale anche per il testo del messaggio. Inoltre fornisce anche la possibilità di controllare i campi "`Received`" per dedurre dai server intermedi da dove è transitato il messaggio. E' previsto l'aggiornamento online e le liste (black/white) possono essere compilate anche dall'utente attraverso il programma. In figura viene riportata la schermata principale.

10.3 MAIL WASHER

Mail Washer è un programma gratuito, in inglese, scaricabile all'indirizzo www.mailwasher.net. Non è solo un ottimo strumento antispam ma è anche un eccellente antivirus. A differenza di molti software antispam, Mail Washer effettua un controllo direttamente sul server di posta prima ancora di scaricare i messaggi. In pratica si collega al provider dove risiede la casella di posta e verifica in modo automatico attraverso opportuni controlli mirati se nella mailbox sono presenti messaggi di spam o virus indicando le email che risultano sospette. In particolare per le e-mail infette da virus si occupa anche della loro eliminazione direttamente sul server evitando in questo modo che raggiungano il computer dell'utente risparmiando anche tempo e denaro per i pesanti allegati a cui si agganciano i worm. Permette di inserire più di un account di posta o semplicemente importarli da altri programmi per la gestione della posta elettronica. Esiste anche la possibilità di scegliere il client da eseguire dopo aver effettuato il controllo della posta con Mailwasher e soprattutto la possibilità di redigere black/white list per eliminare i messaggi provenienti dall'elenco delle liste nere e per indicare come attendibili quelli appartenenti a conoscenti. Inoltre consente di impostare regole per filtrare le email a proprio piacimento secondo caratteristiche determinate dall'utente.

10.4 SAM SPADE

Sam Spade è un software gratuito che richiede la registrazione, è disponibile solo in inglese all'indirizzo http://www.samspade.org/ssw/. Non si tratta come negli altri casi di un programma che utilizza sistemi antispam quali filtri o liste per filtrare la posta ma è una raccolta di tools, utilità di internet che grazie ad un'interfaccia comune consente di salvare i risultati su file di testo. Rappresenta quindi una collezione di strumenti per rintracciare il mittente dei messaggi di posta indesiderati.

Quando si avvia si ha a disposizione un campo di input in cui è possibile digitare un nome di risorsa o indirizzo su cui si possono eseguire determinati comandi. Per esempio, digitando l'indirizzo IP si può ricavare il netblock a cui appartiene oppure digitando un nome si può cercare il corrispondente indirizzo o il whois di dominio e poi sia sui nomi che sugli indirizzi si può effettuare il traceroute. Permette inoltre di esaminare gli header dei messaggi di posta per verificarne la provenienza. Una funzionalità molto importante è quella di poter usufruire di un file di log il cui contenuto riassuma tutte le risultanze dell'indagine in corso e salvarlo su file di testo. Purtroppo l'utility che permette di

cercare se un dato indirizzo IP risulta presente sulle blacklist, è del tutto inutilizzabile in quanto, per il mancato aggiornamento del software, continua ad interrogare solo le blacklist di MAPS che ora non sono più accessibili pubblicamente.

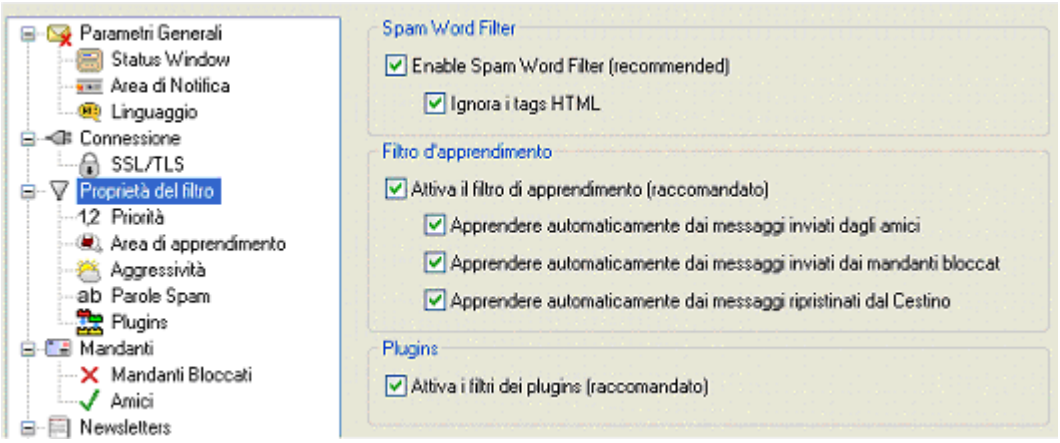
10.5 SPAMPAL

SpamPal è un prodotto freeware open source per sistemi Microsoft che aiuta il client a separare le e-mail legittime da quelle indesiderate.
E' scaricabile all'indirizzo http://www.spampal.org/download.html. SpamPal funziona come un mail proxy trasparente che si posiziona tra il client di posta e la rete e supporta sia il protocollo POP3 che IMAP4. In particolare quando si accorge che un messaggio è indesiderato lo marca nel campo dell'oggetto con la stringa "**SPAM**". Quindi sarà poi il client, attraverso opportune regole che filtrano le e-mail sul campo dell'oggetto, a decidere se eliminare o spostare in una cartella separata lo spam. Può essere considerato un difetto la configurazione del software che è manuale anche se può essere integrata con script aggiuntivi. La filosofia di filtraggio di SpamPal si basa sull'analisi degli header e sulla consultazione delle blacklist dei server DNSBL, ovvero le liste pubbliche di indirizzi IP fonte di spam, che possono essere scelte anche dall'utente, il quale in aggiunta può fornire liste locali personali sia white per i mittenti legittimi che black per quelli irregolari. Con le Ignore list infine è possibile configurare eventuali provider e classi di indirizzi IP che devono sempre essere ritenuti attendibili. La cosa è utile nel caso, non infrequente, un provider sia erroneamente finito in una black list pubblica. SpamPal inoltre è estendibile con utili plugin che aggiungono al filtraggio sull'origine quello sul contenuto. Esiste un filtro che controlla il testo tramite espressioni regolari e uno che filtra i link ai siti degli spammer, già inclusi nel programma mentre si può scaricare l'importantissimo filtro bayesiano.

10.6 SPAMIHILATOR

Spamihilator è un programma antispam freeware, facilmente reperibile e prelevabile all'indirizzo http://www.spamihilator.com/download/.

Per avere la lingua italiana assieme a molte altre è necessario installare un pacchetto aggiuntivo opzionale chiamato language pack. Analogamente a SpamPal, Spamihilator funziona come mail-proxy interfacciando in modo trasparente per l'utente un client di posta e un server POP3 e può utilizzare anche connessioni sicure SSL/TLS. A differenza di Spampal invece questo programma trattiene le e-mail indesiderate in modo tale che non è necessario impostare un filtro anche sul client di posta. Durante l'installazione tramite un wizard si può configurare anche il client di posta con gli account. Utilizza la tecnica del filtraggio tramite filtri bayesiani e quindi è un filtro sul contenuto più che sul mittente. Naturalmente prima di funzionare correttamente deve osservare un periodo di autoapprendimento che viene effettuato dall'utente identificando i messaggi nell'apposita area Apprendimento. Finita questa fase si rivelerà un'ottima arma contro lo spam.
Vengono spostate nel cestino le e-mail riconosciute come spam in modo tale che possano essere poi recuperate nel caso si tratti di falsi positivi. Si possono impostare filtri che riguardano parole prestabilite e soprattutto blacklist che bloccano le email di mittenti illegittimi e whitelist che al contrario accettano traffico senza nessun tipo di controllo sulle persone incluse nella lista, in quanto trattasi di persone conosciute. Si possono bloccare anche gli allegati con estensioni che risultano sospette. Spamihilator è inoltre estendibile grazie ad una serie di utili plugin che permettono di inserire vari tipi di filtri come quelli per la verifica della fonte del messaggio attraverso server DNSBL, che permettono di identificare i link riconducibili a siti di spammer noti e molti altri ancora. Inoltre si possono filtrare immagini, allegati e gestire le newsletters. Per completare è presente una funzione detta parental control che ha lo scopo di proteggere l'accesso con password e una comoda funzione di aggiornamento online. Ecco la schermata principale:



11. SERVER DI POSTA PER LINUX A CONFRONTO

La scelta del server di posta per Linux, ovvero del mail transfer agent più adatto alle proprie esigenze risulta essere molto soggettiva. Inoltre ogni distribuzione Linux installa di default un MTA e quindi spesso si tende a mantenerlo. Le differenze a volte sono anche notevoli e aldilà delle preferenze personali sono proprio quelli che fanno propendere per uno o per l'altro. Fra le caratteristiche più importanti da valutare ci sono modalità d'amministrazione, sicurezza, performance e longevità del prodotto. Normalmente la scelta corre fra le seguenti:

- Sendmail
- Postfix
- Qmail
- Exim

11.1 SENDMAIL

Sendmail è nato agli inizi del 1980, attualmente è utilizzato da quasi i due quinti dei server mail. Dopo aver avuto uno sviluppo lento e poco orientato alla sicurezza l'enorme espansione di Internet l'ha visto in passato protagonista di vulnerabilità. Tuttavia dal 1999 ha subito un notevole sviluppo, fino ad arrivare a metà 2001 in cui si presentava al mondo Opensource in una forma rivoluzionata e con i principi con cui veniva sviluppato che ora comprendevano anche la sicurezza.

Come accennato, Sendmail è stato l'MTA più diffuso nei sistemi UNIX, tanto che anche nelle distribuzioni GNU è stato utilizzato spesso in modo predefinito. Secondo le stime di Sendmail, il programma veicola il 75 per cento della posta scambiata online.

Tuttora è un valido MTA con un gran numero di opportunità di configurazione. Il compito di sendmail è quello di accettare la posta dai Mail User Agent e consegnarla al server di posta appropriato come definito nel suo file di configurazione. Inoltre può accettare connessioni via rete e consegnare i messaggi a caselle di posta locali o ad un altro programma. A seconda delle opzioni con cui viene avviato l'eseguibile `sendmail`, si ottiene un demone in ascolto della porta SMTP (25), oppure si ottiene la trasmissione di un messaggio fornito attraverso lo standard input, oppure si hanno altre funzioni accessorie. Solitamente, Sendmail viene distribuito già configurato in modo

standard, sia per l'attivazione del servizio SMTP, sia per la gestione della trasmissione dei messaggi e della consegna locale.

In particolare Sendmail invia un messaggio a uno o più destinatari, occupandosi di instradarlo e quindi recapitarlo al corretto indirizzo. Non ha lo scopo di fornire un'interfaccia user-fiendly; infatti a questo pensano altri programmi, Sendmail è usato solo per consegnare messaggi a tracciato prestabilito. Non ha flags, legge i suoi standard input fino ad un end-of-file oppure ad una linea contenente un solo punto e manda una copia del messaggio a tutti gli indirizzi presenti nella lista. Determina l'instradamento basandosi sulla sintassi e il contenuto degli indirizzi.

Generalmente su gran parte delle distribuzioni Linux, Sendmail viene installato di default e in molti casi viene eseguito all'avvio per restare in ascolto sulla porta 25 del localhost.
In questo modo viene usato per inviare la posta locale, ma non rimane in ascolto su un indirizzo pubblico. Il pregio di Sendmail è la sua estrema configurabilità. Il suo difetto è lo stesso pregio: l'estrema configurabilità implica un'estrema complessità. La sua sintassi infatti è ridotta ai minimi termini

Questo è il motivo che spesso tende a far ricadere la scelta su un altro mail server.

Recentemente è stata scoperta una falla nella sicurezza. Si tratta di un problema particolarmente pericoloso in quanto per iniziare l'attacco basterebbe un semplice messaggio di posta elettronica e chiunque fosse in grado di sfruttare la falla potrebbe intercettare le e-mail, leggerle, modificarle o interromperne il flusso. Ad essere colpite dalla vulnerabilità risultano tutte le versioni commerciali di Sendmail, così come le versioni open source a partire dalla 5.79. Sul sito del software sono disponibili le ultime patch per risolvere il problema. Tuttavia questo è stato visto come il colpo di grazia per il declino di sendmail a beneficio degli altri MTA. Sendmail era da sempre considerato uno dei simboli dell'OpenSource e di Internet stessa; ha consegnato messaggi fin dagli albori della rete, scambiando email e file fra server UNIX quando ancora non ci si preoccupava molto della sicurezza dei sistemi e dei protocolli. E' anche cresciuto di versione in versione ma nonostante tutto questo si è portato dietro una logica complessa, estremamente flessibile ma a suo modo contorta, che ha richiesto ripetute correzioni e pezze per rincorrere le sempre più variegate vulnerabilità che gli venivano trovate. Negli ultimi tempi il suo sviluppo si era consolidato, da almeno un paio d'anni non venivano scoperti bug pericolosi e, nonostante la crescente concorrenza di altri mailer OpenSource come Postfix e Qmail, nati dopo e intrinsecamente più sicuri, è stato per molti una scelta stabile e affidabile. Sendmail si comporta senza ombra di dubbio in modo eccellente in tante occasioni ma per esempio per le connessioni dial-up esistono software che lo superano in semplicità di utilizzo, sicurezza ed efficienza.

11.2 POSTFIX

Postfix è stato scritto inizialmente da Wietse Venema, esperto di sicurezza autore e coautore di diversi e famosi tools di controllo e monitoraggio. E' attualmente disponibile sotto licenza IBM. Postfix è il più giovane degli MTA qua elencati, tuttavia ha presentato fin da subito enormi potenzialità. Come funzionalità si colloca fra Exim e Qmail, è dotato di un file monolitico di configurazione ma il corpo del programma è stato "spezzato" in modo che un demone svolgesse un'unica funzione, prevenendo così l'acquisizione tramite scalata di diritti root tramite vulnerabilità. Postfix è stato studiato anche per reggere grossi volumi di lavoro ed è, come Exim, un rimpiazzo di sendmail.

Gli scopi principali del progetto che ha portato alla creazione di Postfix sono l'ampia diffusione, la compatibilità, le performance, la robustezza, la sicurezza e la flessibilità. Per raggiungerli si è puntato sulle buone caratteristiche del codice e la compatibilità soprattutto con Sendmail, di cui si vuole che sia il degno sostituto. La migrazione fra i due programmi è stata resa facile, Postfix dall'esterno si comporta come Sendmail, ma internamente è completamente diverso. Postfix, infatti, può gestire i messaggi ed alcuni file di configurazione (come ad esempio `/etc/aliases`) con lo stesso formato utilizzato da Sendmail e nello stesso tempo si svincola completamente dal complicato file di configurazione `sendmail.cf`. Dal punto di vista prestazionale Postfix è considerato fino a tre più veloce del suo avversario ed è inoltre capace di inviare e ricevere un milione di diversi messaggi al giorno. Per ottenere ciò sono state adottate tecniche per ridurre l'overhead nella creazione dei processi e nella gestione dei file. Per quanto riguarda la robustezza è previsto che, sotto pesanti carichi, i programmi che compongono Postfix rallentino per mantenere stabile il sistema. Infatti Postfix è costituito da numerosi programmi che cooperano, ciascuno con un compito ben preciso e questo lo rende molto flessibile in quanto i vari programmi possono essere attivati, disattivati, sostituiti e modificati in base alle esigenze specifiche.

Postfix supporta entrambi i formati mailbox e maildir, il primo tipico di Sendmail, il secondo di Qmail. Il sistema a mailbox utilizza un unico file per conservare tutti i messaggi di una mailbox (tipicamente `/var/spool/mail` o `/var/mail`). Nel sistema maildir al contrario ciascun messaggio è memorizzato in un file e ciascuna mailbox è composta da una directory collocata nella home dell'utente. In caso ci fosse l'esigenza di usare partizioni montate in NFS il formato maildir è

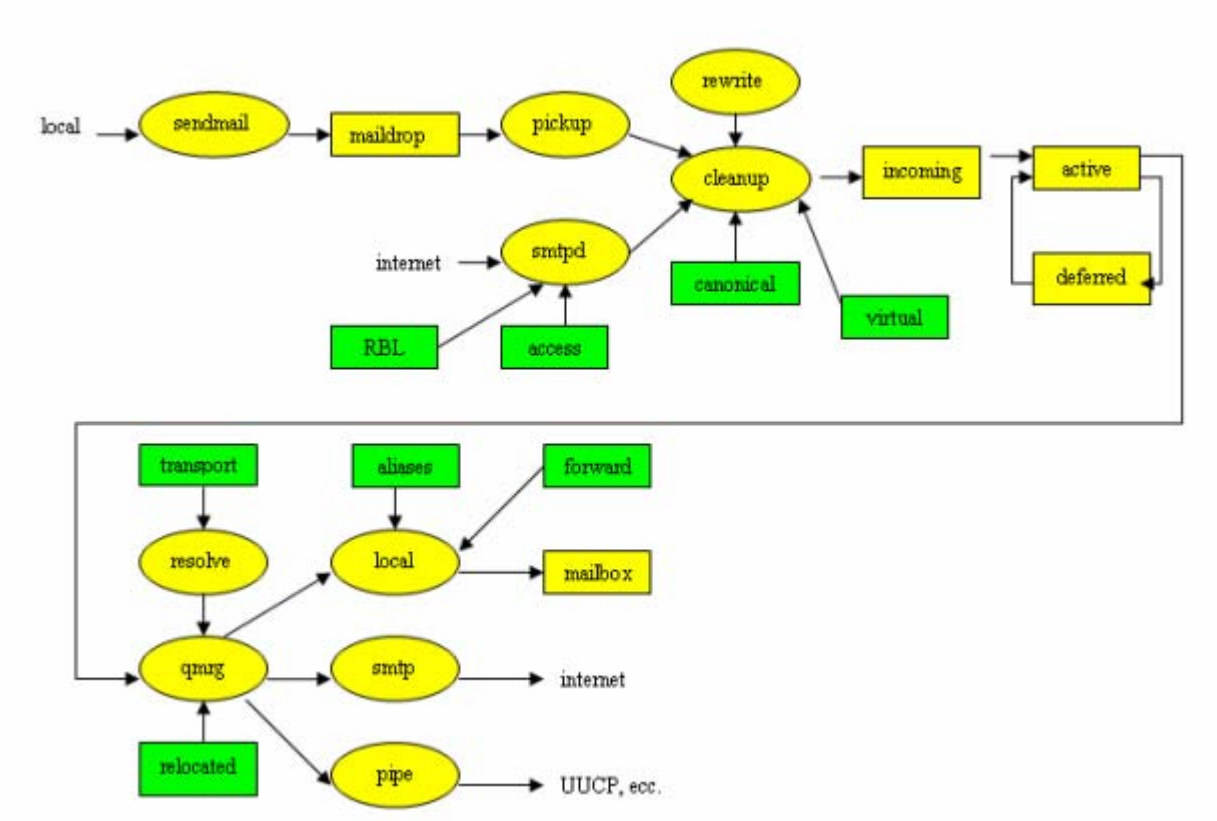
obbligatorio. Inoltre Postfix supporta molti formati per le tabelle usate nella gestione dei dati (lookup tables), come ad esempio alias, domini locali, reti e client autorizzati. Sono presenti numerose possibilità di filtraggio dei messaggi sulla base del mittente, del destinatario, dell'intestazione dei messaggi e del loro contenuto. Si possono poi intraprendere determinate azioni sui messaggi filtrati. Esiste anche la possibilità di attivare il servizio di posta attraverso un canale crittografato. Tutte le configurazioni avvengono editando semplici file di testo.

Postfix, come tutti i programmi, presenta anche alcuni svantaggi nei conforti di altri mail server: la mancanza di strumenti dedicati ed integrati come ad esempio server POP3 e IMAP, un sistema per l'autorizzazione al relay degli utenti con IP dinamico (dial-up) chiamato anche "pop-before-smtp". Risulta quindi necessario integrare Postfix con altre componenti esterne.

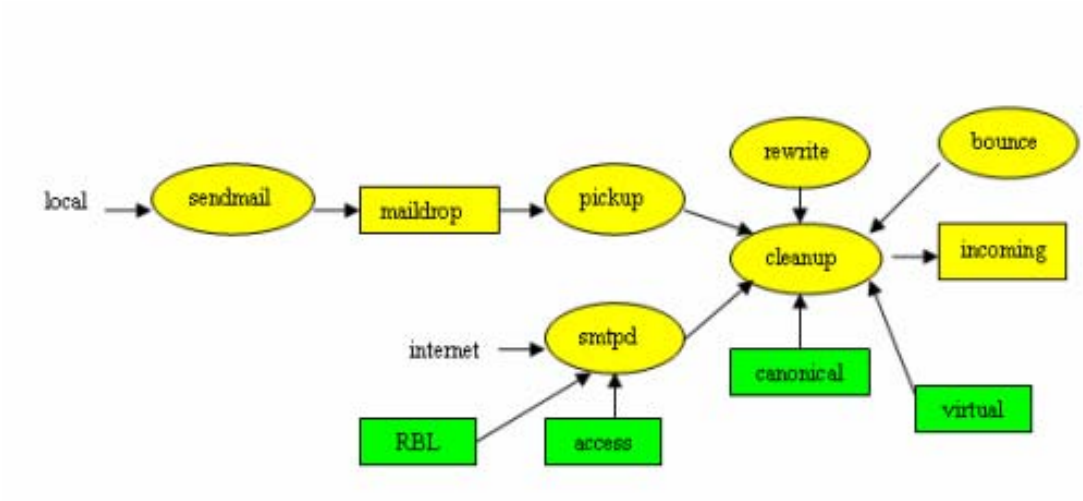
Architettura

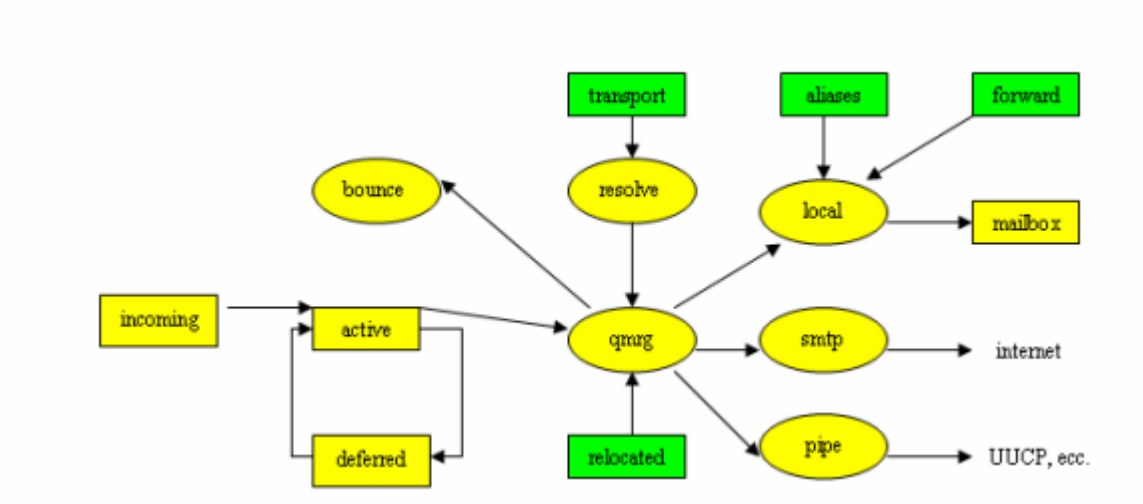
I programmi sono richiamati dal demone principale `master` che gestisce tutti gli altri programmi e rimane sempre attivo. La ricezione dei messaggi avviene ad opera di due demoni in base alla loro provenienza: esterna (processo `smtpd` che gestisce l'apertura di nuove connessioni SMTP) o interna (processo `pickup` che si occupa di accettare i messaggi locali). L'input dei dati viene gestito dal processo `cleanup`. Postfix si avvale di quattro tipi di code per la gestione dei messaggi: `maildrop`, `incoming`, `active`, `deferred`. La prima è la coda della posta in arrivo; i messaggi poi passano dopo essere stati elaborati alla coda incoming. `deferred` invece è la coda dei messaggi che non sono stati inviati mentre `active` è la coda per i messaggi di ridotte dimensioni per non caricare troppo il server. Il programma `qmgr` è il gestore delle code e fornisce le informazioni necessarie ai demoni `local`, `smtp` o `pipe`. `Local` si occupa della consegna locale dei messaggi, `smtp` di quella remota mentre tramite `pipe` è possibile inviare i messaggi a comandi esterni. `Trivial-rewrite` risolve l'indirizzo della destinazione, `bounce` genera un messaggio di avvertimento se un messaggio non può essere inviato. Infine completano il pacchetto alcuni programmi di amministrazione.

Vediamo ora uno schema riassuntivo che ci permette meglio di capire il funzionamento di postfix, dove gli ellissi gialli indicano i programmi, i rettangoli gialli sono le code dei messaggi e i file contenenti i messaggi e infine i rettangoli verdi sono le tabelle di lookup. Il demone principale `master` che gestisce tutti i programmi non è rappresentato.



In questa figura invece focalizziamo l'attenzione sulla gestione della ricezione della posta e nella seguente la gestione dell'invio della posta.





Caratteristiche

L'alta sicurezza, la facilità di installazione e configurazione e le ottime prestazioni che potrebbero far preferire Postfix agli altri MTA sono ottenute grazie a determinate caratteristiche che si suddividono in diversi "strati" di protezione. Andiamo ora ad analizzarli uno per uno:

- Bassi privilegi: tutti i processi tranne `master` girano senza i privilegi di root.
- Gabbia chroot: si tratta di una porzione ristretta del filesystem in cui possono essere eseguiti tutti i programmi che compongono Postfix, in modo da introdurre una barriera di difesa anche se non è una garanzia di assoluta sicurezza.
- Isolamento: Postfix è stato suddiviso in più programmi che svolgono determinate attività in modo tale che nel caso di violazione del sistema sia più difficile aver accesso ai dati della macchina.
- Ambiente controllato: il demone `master` ha l'unica funzione di controllare gli altri processi in modo da non creare relazioni padre-figlio con processi utente.
- Set-uid: Postfix non utilizza programmi che girano con il bit setuid attivo, ovvero programmi che girano con i privilegi dell'utente proprietario del file, indifferentemente dall'utente che li ha invocati.
- Fiducia: la filosofia di Postfix è quella di non considerare affidabile nessun contenuto a maggior ragione i dati provenienti dalla rete.

- Controlli sull'input: l'allocazione in memoria è fatta dinamicamente, le linee lunghe dei messaggi vengono suddivise e ricostruite prima dell'invio.

Approfondiremo meglio i dettagli sull'installazione e configurazione di Postfix nella parte pratica che si occuperà proprio di questo, mentre tralasceremo quelli relativi agli altri MTA.

11. 3 QMAIL

Qmail è un MTA per sistemi UNIX e LINUX. Il pacchetto `qmail` è stato scritto da D. J. Bernstein. E' stato pensato per sostituire Sendmail a causa delle varie falle di sicurezza che minavano la stabilità del programma. Usa SMTP per scambiare messaggi con gli altri MTA su altri sistemi e presenta una separazione delle funzioni in più demoni. Non causa gravi traumi agli utenti e agli amministratori che usavano sendmail. Dispone, infatti, di un sendmail "wrapper" che permette di continuare ad usare tutti gli altri programmi che abbiano necessità di sendmail come se nulla fosse cambiato. Un discreto numero di mailserver si basano su questo prodotto, principalmente realizzato con particolare cura alla sicurezza, realizzato mediante un corpo principale di gestione della posta e tanti piccoli programmi userspace per poterlo gestire. Qmail ha una considerevole cura di path o add-on inviati dai contributors e reperibili direttamente sul sito ufficiale. Viene utilizzato sempre più frequentemente rispetto a Sendmail grazie alle sue notevoli prestazioni e alla sicurezza; esiste infatti un premio di 1000 dollari per chi riesce a trovare un "buco" che renda possibile provocare danni al server su cui gira.

Vediamo i principali vantaggi di Qmail rispetto agli altri Mail Transfer Agent:

- Sicurezza: la sicurezza non è un obiettivo, ma un requisito assoluto. L'invio di posta è critico per gli utenti; non può essere disabilitato, perciò deve essere completamente sicuro. Qmail è stato progettato per un alto standard di sicurezza anche per rimpiazzare Sendmail e le sue lacune riguardo questo tema. Si è cercato di limitare al massimo il codice che gira sotto i privilegi di root: solo `qmail-start` e `qmail-Ispawn` necessitano dei privilegi di root e solo `qmail-queue` (ovvero il demone che permette di aggiungere un messaggio alla posta che si trova già in coda) è setuid. Al contrario Sendmail è costituito da circa 46000 righe di codice tutte eseguite con i privilegi di root.

- Performance: Anche se gli invii remoti sono intrinsecamente limitati dalla lentezza del DNS e dell'SMTP, Qmail parallelizza la consegna della posta, effettuando sino a 20 consegne simultanee per default permettendo una velocità elevata nelle mailing list. E' efficiente nel trasferire grandi quantitativi di posta. Non a caso viene utilizzato da grossi fornitori di caselle di posta elettronica gratuite e anche per gestire mailing list ad elevato traffico. Inoltre è indicato per i server che hanno limitate risorse in quanto consente di risparmiarne molte.
- Affidabilità: E' a prova di crash del sistema. Una volta accettato il messaggio, Qmail garantisce che non vada perduto seguendo la filosofia di salvataggio "straight-paper-path". Inoltre supporta il formato maildir, nuovo e super affidabile. I maildir, a differenza dei file mbox o delle directory mh, non vengono corrotti se il sistema va in crash durante l'invio. Meglio ancora, non solo un utente può tranquillamente leggere la propria posta attraverso NFS, ma qualsiasi numero di client possono inviargli posta allo stesso tempo.
- Semplicità: E' piccolo rispetto agli altri MTA ma allo stesso tempo ha le stesse caratteristiche. E' molto più leggero degli altri MTA per almeno tre differenti motivazioni:
    1. gli altri MTA hanno meccanismi separati per il forwarding, l'aliasing e le mailing list. Qmail ha un semplice meccanismo di inoltro che permette agli utenti di gestire le proprie mailing list.
    2. Gli altri MTA offrono diverse modalità di invio, da quella veloce ma insicura a quella più lenta con accodamento dei messaggi. Qmail-send è richiamato istantaneamente dai nuovi elementi in coda, cosicché qmail ha un'unica modalità di invio: veloce e con accodamento messaggi.
    3. Gli altri MTA includono una versione specializzata di inetd (demone, servizio che ascolta sulle porte specificate nel suo file di configurazione e fa avviare il relativo servizio nel momento in cui viene fatta una richiesta.) che controlla il carico medio. L'architettura di qmail limita intrinsecamente il carico della macchina, perciò `qmail-smtpd` può tranquillamente funzionare utilizzando l'inetd del sistema.
    4. Inoltre è facile da installare e da configurare.
- Flessibilità: Qmail è molto flessibile grazie alla possibilità di aggiungere piccoli add-on (estensioni) che gli permettono di fare tutto ciò che fa sendmail, ma senza richiedere conoscenze particolari: tutta la configurazione si esegue editando semplici file di testo ben commentati.

11.4 EXIM

Exim è nato partendo dal codice di Smail3 (nato a sua volta come derivato da sendmail per i problemi di sicurezza di sendmail stesso). E' stato concepito come general purpose mailer, senza una rivoluzione concettuale della struttura come è accaduto a Qmail. E' dotato di un file di configurazione monolitico, tramite il quale si può facilmente controllare il demone in tutti i suoi aspetti. Finora non ha mostrato un discreto interesse per i temi della sicurezza e nemmeno vulnerabilità tali da compromettere l'integrità dei server su cui era installato. Exim consegna la posta sia in locale che in remoto e può accettare numerosi plugin tutti richiamabili dal file di configurazione monolitico. Sta crescendo in popolarità in quanto è open source, scalabile e ricco di caratteristiche come le seguenti:

- Compatibilità con le interfacce e le opzioni di Sendmail (per questo Exim è utilizzato anche per rimpiazzare Sendmail);
- Sofisticato trattamento degli errori;
- Supporto per l'analisi di molti tipi di indirizzi, incluse espressioni regolari;
- Innumerevoli parametri per migliorare le prestazioni e trattamento di enormi volumi di email.

Exim è quindi un MTA che offre qualche piccolo vantaggio rispetto a Sendmail: ha un sistema di configurazione meno complesso, è predisposto per IPv6 e quando possibile utilizza i processi senza i privilegi dell'utente root, in modo da lasciare meno occasioni alle aggressioni. Nonostante ciò è abbastanza compatibile con l'uso di Sendmail. Questa compatibilità riguarda tre punti fondamentali: il file `/etc/aliases`, i file `~/.forward` e un collegamento che simula la presenza dell'eseguibile `sendmail`. Inoltre l'eseguibile `exim` accetta buona parte delle opzioni standard di sendmail. La sintassi e il comportamento di Exim possono risultare leggermente più complesse per particolari aree come il filtraggio, l'hosting virtuale e le risposte automatiche.

Vediamo ora uno schema riassuntivo che mette a confronto i server di posta.

	Exim	**Postfix**	**Sendmail**	**Qmail**
Sicurezza	Medio-bassa	Alta	Bassa	Molto alta
Installazione	Media	Medio-facile	Facile	Medio-difficile
Configurazione	Medio-facile	Facile	Difficile	Facile
Performance	Medie	Alte	Medie	Alte
Maturità	Bassa	Media	Alta	Media
Documentazione	Molta	Media-molta	Molta	Molta
Features	Medie-molte	Medie	Molte	Poche*
* disponibili patch per nuove funzionalità, necessitano ricompilazione				

12. CLIENT DI POSTA PER LINUX A CONFRONTO

Attualmente per Linux esistono diversi programmi che permettono di ricevere, leggere ed inviare posta elettronica, naturalmente ognuno di essi ha i suoi pregi e difetti. La scelta del client di posta deve essere quindi oculata in modo che possa soddisfare tutte le esigenze dell'utente e, al tempo stesso, garantisca velocità e affidabilità. Le differenze principali riguardano l'interfaccia, l'integrazione con l'ambiente grafico e l'utilizzo di risorse necessarie al funzionamento del programma. Passiamo ora in rassegna i principali e più diffusi client di posta ovvero i Mail User Agent:

- KMail
- Mozilla Thunderbird
- Evolution
- Sylpheed
- Pine
- Mutt

12.1 KMAIL

KMail è il client di posta di default di KDE. Il grande pregio di un software per client di posta come KMail, che gli consente di avvantaggiarsi di un'elevata stabilità, è quello di concentrarsi sulle funzioni che sono ritenute fondamentali tralasciando quelle che poco hanno a che fare con la posta, e sull'integrazione con l'ambiente grafico. Elenchiamo le caratteristiche principali: supporto ai server POP, IMAP e SMTP con possibilità di effettuare connessioni sicure utilizzando i protocolli SSL o TLS, autenticandosi in modo da essere autorizzati per il relay verso l'esterno. Inoltre fornisce la possibilità di importare i messaggi di posta elettronica già scaricati su altri client anche di Windows come Microsoft Outlook e Pegasus Mail. Supporta i formati di memorizzazione Maildir e Mailbox e i sistemi di cifratura come PGP e GPG per poter gestire le e-mail crittografate. Consente di scaricare le e-mail da più account anche di server diversi. E' possibile impostare alcuni filtri per indirizzare la posta in arrivo in varie cartelle analizzando le intestazioni. Non è ancora stato integrato un sistema antispam ma si può redirigere la posta ad un programma esterno che si occupi di filtrarla. Già la prossima versione comunque comprenderà una funzionalità per filtrare la posta indesiderata, oltre a migliorare il supporto S/MIME. Una caratteristica molto particolare offerta da questo client è la possibilità di comporre un messaggio con un editor di testo alternativo a quello fornito e l'elevato numero di dettagli che si possono inserire nella rubrica per i propri contatti.

Da notare un piccolo difetto: manca un vero e proprio wizard (ovvero un aiuto interattivo che guida l'utente fino al compimento di una determinata azione) per configurare il programma e inoltre le varie impostazioni personali vanno configurate in modalità manuale.

12.2 MOZILLA THUNDERBIRD

Mozilla Thunderbird è nato come un progetto di Mozilla con lo scopo di creare un client di posta che si integra con il browser web. Ha un'interfaccia estremamente configurabile. Come per KMail è possibile importare la posta sia da file mailbox che dai client di posta principali. Naturalmente sono supportate le connessioni POP, IMAP e SMTP e la possibilità di usare SSL o TLS per le connessioni sicure e PGP/GPG per quelle cifrate. Comprende un wizard per la configurazione della posta elettronica anche se di un solo account, per gli altri è necessario procedere manualmente e con la versione 0.8 è possibile unire tutti i messaggi provenienti da account diversi in un'unica cartella di posta in arrivo. Molto utile la funzionalità che permette di estendere il programma con plugin esterni. Oltre alla possibilità di filtrare i messaggi di posta per suddividerli in cartelle diverse offre soprattutto la sempre più importate funzione di filtrare lo spam adottando filtri che si adattano in base alle impostazioni dell'utente. La posta scartata può essere eliminata o spostata in una cartella apposita in modo da evitare di cancellare falsi positivi. La rubrica molto fornita consente anche l'accesso via LDAP ad un repository centrale. Da notare la portabilità del programma: è infatti possibile eseguirlo su piattaforme diverse tra cui Windows e Mac, oltre a Linux. Questo ha un notevole vantaggio in quanto è possibile condividere la posta per esempio su computer dual boot.

12.3 EVOLUTION

Evolution è un progetto della Free Software House Ximian. Rappresenta forse il client di posta per Linux più completo. Offre funzionalità ulteriori come l'agenda e la rubrica condivisa con possibilità di sfruttare un database LDAP ed è un client grafico integrato all'interno dell'ambiente grafico GNOME, scritto utilizzando le librerie GTK. Come gli altri anche questo programma supporta SMTP, POP, IMAP, SSL, PGP/GNUPG e server con autenticazione. Inoltre è possibile essere autorizzati a spedire posta tramite un server SMTP utilizzando le procedure di autenticazione di un server POP, ovvero con la modalità "POP before SMTP". Dispone di filtri per la suddivisione dei messaggi in cartelle che si avvalgono di vari criteri, con anche la possibilità di passare le e-mail ad un programma esterno per poi eseguire determinate azioni con il valore di ritorno di quel determinato programma. Oltre a mailbox e maildir Evolution può importare la posta di Eudora, Netscape, Mozilla o Outlook Express. All'avvio del programma parte la configurazione con un wizard che permette di impostare più account. Una caratteristica che lo distingue dagli altri è

rappresentata dalla possibilità di organizzare i messaggi in cartelle virtuali (virtual folders) senza spostarli dalla posizione originale. Manca la funzionalità di filtrare lo spam (che sarà introdotta con la nuova versione) che è sopperita dalla possibilità di integrarsi con un programma esterno.

12.4 SYLPHEED

Sylpheed è un programma estremamente funzionale, leggero e stabile, è scritto utilizzando le librerie GTK ed è studiato per sistemi con poche risorse. Come tutti consente connessioni con POP, IMAP, SMTP anche utilizzando SSL, PGP/GPG ma non S/MIME. All'avvio parte il menù di configurazione che permette anche di importare messaggi organizzati in mailbox e di creare più account. Non dispone ancora di filtri antispam ma solo di quelli che filtrano la posta per suddividerla in cartelle diverse. Consente la connessione ad un server LDAP anche se la rubrica è piuttosto essenziale.

12.5 PINE

Pine è un progetto dell'Università di Washington ed è un client testuale semplice che grazie ai menù con varie opzioni permette di gestire la posta in modo molto efficace. Differentemente dagli altri client che sono rilasciati con licenza GPL, Pine è fornito con licenza OpenSource. Per scaricare la posta non è autonomo ma si deve appoggiare a programmi esterni e per l'invio non è necessario installare un MTA sulla propria distribuzione. Supporta il formato mailbox e tramite patch esterne anche maildir ma non connessioni sicure come SSL e nemmeno PGP o GPG e questo rappresenta un notevole limite. Inoltre è l'unico client che non consente la visualizzazione dei thread (sequenza di risposte ad un messaggio iniziale). E' fornito di una buona rubrica che dà la possibilità di accedere a LDAP esterni. Non sono compresi sistemi antispam anche se è possibile usare programmi esterni.

12.6 MUTT

Mutt rappresenta il client di posta testuale più evoluto, ha grande stabilità e modeste pretese di risorse. A differenza di Pine, supporta la visualizzazione dei messaggi per thread. Include il supporto a PGP/GPG, server POP, IMAP e SSL. Tuttavia è utilizzato spesso appoggiandosi a programmi per la consegna in locale della posta. E' altamente configurabile, ogni azione è associata ad un tasto. Mutt non è in grado di spedire messaggi autonomamente e quindi è necessaria la presenza di un MTA sul proprio computer.

Riporto ora uno schemino riassuntivo:

Client	POP	IMAP	SMTP	SSL	Multi Account	PGP GPG	S/MIME	Filtri	Anti Spam	LDAP	Caratteristiche
Evolution	Si	Si	Si	Si	Si	Si	No	Si	No	Si	Virtual folders, Rubrica avanzata, Agenda
KMail	Si	Si	Si	Si	Si	Si	No	Si	No	Si	Virtual folders, Rubrica avanzata, Possibilità di usare editor esterno
Thunderbird	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Estendibile con plugin, Multipiattaforma
Sylpheed	Si	Si	Si	Si	Si	Si	No	Si	No	Si	Client molto leggero
Pine	No	Si	Si	No	No	No	No	No	No	Si	Client testuale, richiede che la posta sia scaricata con un programma esterno, non supporta i thread
Mutt	Si	Si	No	Si	No	Si	Si	No	No	Si	Client testuale, supporta i thread, configurabile e personalizzabile, richiede un MTA installato

13. NUOVI SVILUPPI

La battaglia contro lo spam vede emergere una nuova importante iniziativa, oltre al costante affinamento dei filtri antispam. E' da segnalare, infatti, una nuovissima tecnica che dovrebbe contribuire ad identificare indirizzi e-mail fasulli dai quali proviene lo spamming: DKIM (Domain Keys Identified Mail). E' infatti in grado di impedire agli spammer di dissimulare la loro identità durante la spedizione di messaggi indesiderati. Si tratta di una specifica, un nuovo standard di autenticazione che Cisco, Microsoft, IBM, Yahoo, Sendmail, America Online, EarthLink, VeriSign e Pgp hanno presentato all'IETF (Internet Engineering Task Force) affinchè sia preso in considerazione come standard per il settore delle e-mail e per favorire l'adozione su larga scala di questa tecnologia. Le discussioni su questo tema faranno parte del 63esimo meeting dell'IETF avvenuto il 31 luglio 2005. Sebbene la lotta contro lo spam accomuna tutti, è stato molto difficile raggruppare tutti sotto uno stesso standard. Il primo tentativo si deve a Microsoft che proponeva un `Sender ID` ma purtroppo non diede gli esiti sperati e la IETF non appoggiò il progetto. DKIM ha il vantaggio di essere completamente "royalty-free".

DKIM usa una chiave pubblica crittografica allegando ai messaggi di posta in uscita una firma elettronica che consente ai destinatari di verificare l'identità del mittente. Questa firma, infatti è accoppiata ad una firma sul server che invia il messaggio permettendo al server del destinatario di bloccare i messaggi la cui firma non corrisponde al server del mittente. In questo modo i messaggi vengono marcati fin dall'origine fornendo così un'ottima soluzione agli attacchi di spam e phishing (uso illegale di sistemi quali e-mail e siti web in maniera tale da simulare raccolte di dati da parte di importanti compagnie presenti su Internet, con lo scopo di carpire informazioni, numero di carta di credito in primo luogo). DKIM è stato sviluppato per fornire strumenti potenti ed accurati per garantire un livello di protezione contro la contraffazione degli indirizzi e-mail, identificando i messaggi legittimi. Protegge quindi anche il marchio delle aziende che fondano la loro attività sulle transazioni economiche.

Le specifiche combinano gli elementi principali delle tecnologie di Yahoo (Domain Keys Technology) e di Cisco (Internet Identified Mail Technology), creando una sinergia dalle ottime potenzialità.

DKIM ha richiesto circa un anno di lavoro. E' da agosto 2004, infatti, che queste compagnie si sono impegnate per collaborare a collaborare per il nuovo metodo di Email Authentication. E' previsto inoltre il rilascio dello standard in maniera del tutto "free" e probabilmente anche alla comunità open-source.

Purtroppo ho potuto solo accennare questa innovativa tecnica in quanto non esiste ancora molta documentazione anche perché è appena uscita e non è ancora stata ratificata a standard. DKIM,

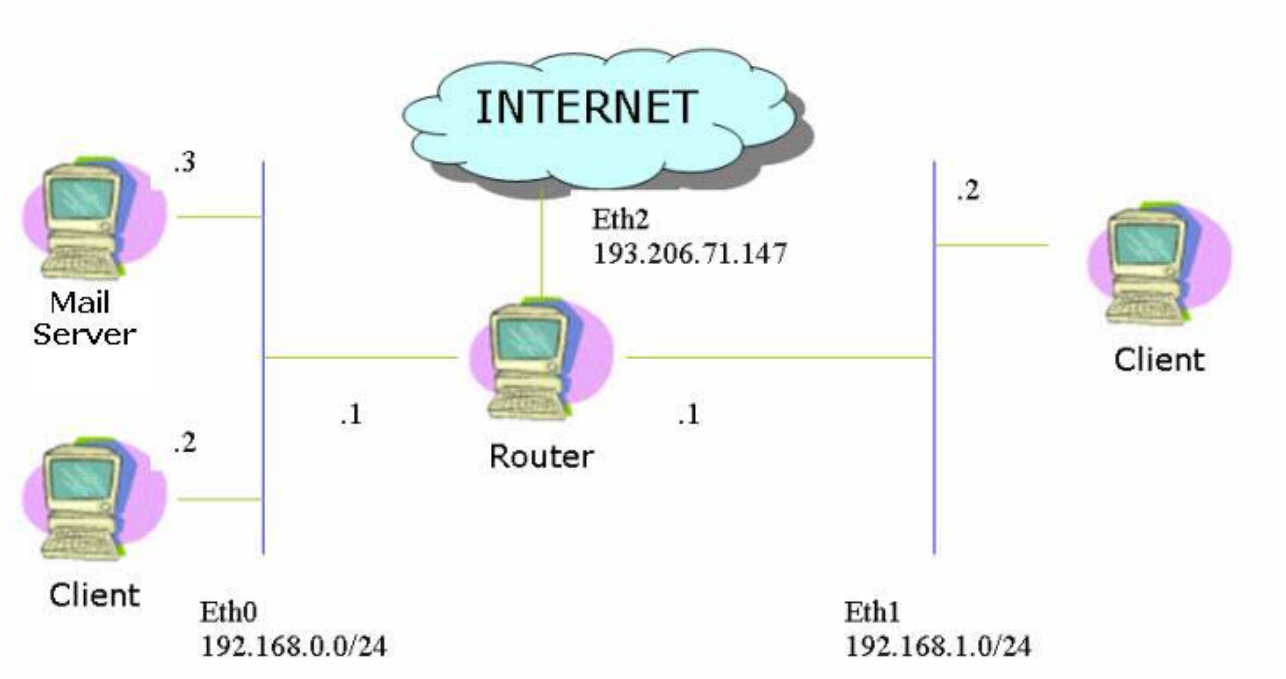
infatti, al momento è disponibile sul sito www.ietf.org solo come IETF Internet Draf ed in particolare si può trovare ai seguenti link:

- ✓ http://www.ietf.org/internet-drafts/draft-allman-dkim-base-00.txt
- ✓ http://mipassoc.org/dkim/specs/draft-allman-dkim-base-00-10dc.html

14. SPERIMENTAZIONE PRESSO IL LABORATORIO RETI

In questa parte della trattazione andrò a descrivere in dettaglio come implementare un server di posta con filtri antispam. In particolare ho scelto di installare e configurare come MTA Postfix e come software antispam SpamAssassin e di integrarli. Utilizzo Postfix in quanto prima di tutto è già fornito con la distribuzione Linux che adotterò per realizzare il mail server, ovvero Fedora. Inoltre è facile da configurare, estremamente efficiente nonché molto flessibile e sicuro, grazie anche alla sua struttura modulare che già abbiamo analizzato. Anche SpamAssassin è già incluso in Fedora e inoltre l'ho scelto in quanto è uno dei pochi che permette di sfruttare l'enorme potenziale dei filtri antispam sopratutto quelli bayesiani. Infine come client di posta ho utilizzato KMail.

Illustro ora brevemente con una figura la configurazione della rete che ho utilizzato. Il mail server è stato configurato sulla macchina avente indirizzo privato 192.168.0.3 con hostname labreti4. E' stato utilizzato un solo dominio (labreti4.it) e l'accesso è stato consentito ai soli appartenenti alla rete aziendale ovvero le sottoreti 192.168.0.0/24 e 192.168.1.0/24.



Per consentire l'inoltro delle e-mail ho dovuto editare il file `/etc/hosts` andando ad inserire su tutte le macchine la seguente un modo da poter riconoscere il mail server:

```
196.168.0.3     labreti4.it     labreti4
```

14.1 MTA: POSTIX

14.1.1 Installazione

Come già detto, non ho dovuto provvedere all'installazione di Postfix, tuttavia per chiarezza spiegherò brevemente come avviene.

Innanzitutto se è configurato come MTA principale Sendmail, prima di effettuare la migrazione è necessario fare il backup di tutti i files di configurazione copiandoli in una cartella adatta allo scopo. In questo modo nel caso qualcosa andasse storto si può ripristinare il sistema senza grossi problemi. In particolare è consigliato copiare i file contenuti in `/etc/` oppure `/etc/mail/` fra cui `aliases`, `access`, `localhost-names`, `virtusertable`, `mailertable`, `domaintable`. Soprattutto è necessario conservare il binario `sendmail`, che si può trovare lanciando dal terminale il comando `which sendmail` (di solito si trova in `/usr/bin/`).

Si può procedere ora con l'installazione. Esistono due modi: utilizzare pacchetti già compilati per la nostra distribuzione oppure partire direttamente dai sorgenti. Il primo caso è sicuramente il più agevole in quanto non è necessaria la compilazione; l'unico svantaggio è quello di non poter disporre di un binario personalizzato. Tuttavia questa soluzione è la più conveniente e si adatta alla maggior parte delle situazioni. Si possono quindi usare i pacchetti precompilati disponibili ormai in molte distribuzioni oppure scaricare postfix direttamente dal sito originale www.postfix.org e scegliere nell'area "Packages and Ports" i binari precompilati o il port. Non resta che installare il pacchetto lanciando il comando:

```
# rpm –ivh postfix-versione.i386.rpm
```

Se la scelta ricade sui sorgenti, sempre nel sito di postfix nell'area "Download" sul mirror preferito si possono scaricare. Si procede quindi con la compilazione. Decomprimiamo e scompattiamo l'archivio dei sorgenti

```
# tar xvzf postfix-versione.tar.gz
# cd postfix-versione/
```

Si può quindi dare un'occhiata al contenuto della cartella e in particolare al file `INSTALL`, alle cartelle `html/` e `README_FILES/` che contengono documentazione e alla cartella `config/` che contiene esempi di configurazione.

Passiamo ora alla fase di compilazione vera e propria lanciando `make`. E' necessario poi creare il gruppo `postfix` e un utente `postfix` che non siano utilizzati da nessun account. In particolare l'utente `postfix` dovrà appartenere al gruppo `postfix` e inoltre dovrà essere senza shell, senza home e senza possibilità di login. Creiamo anche il gruppo `postdrop` verificando sempre che non

sia già in uso. Procediamo quindi con `make install` e se non abbiamo ricevuto messaggi di errore ci verranno poste alcune domande. Se le nostre esigenze non sono specifiche e particolari si può benissimo accettare i valori di default proposti.

Ecco il listato dei comandi:

```
# make
# adduser postfix -s /bin/false -d /dev/null
# passwd -l postfix
# groupadd postfix
# groupadd postdrop
# make install
```

Ora l'installazione è completata.

14.1. 2 Configurazione di base

Nella directory `/etc/postfix` si potranno trovare tutti i file di configurazione e le tabelle. Passiamo ora in rassegna i principali file di configurazione:

- ✓ `access`: in questo file si stabiliscono diritti di accesso selettivi per l'accettazione di e-mail, tenendo conto di mittente, destinatario e hostname.
- ✓ `aliases`: serve per reindirizzare e inoltrare le mailbox locali. Ogni volta che questo file viene modificato è necessario lanciare il comando `newaliases` perché le modifiche abbiano effetto.
- ✓ `canonical`: trascrive indirizzi di posta e nomi di dominio per mittenti e destinatari, serve spesso a mascherare indirizzi di posta e domini interni.
- ✓ `main.cf`: è il file di configurazione principale in cui vengono stabiliti tutti i parametri fondamentali di Postfix. Attraverso questo file è possibile configurare Postfix come un semplice relay server per una piccola rete casalinga fino ad arrivare a configurazioni di alto livello per sistemi "enterprise". Il file si presenta molto ben commentato e spesso i valori di default vanno benissimo, ma rimane un file complesso e molto grosso, i diversi parametri sono molti, più di 250 stando a quanto scritto nelle primissime righe di commento. Per questo generalmente Postfix dispone di un nutrito set di file di esempio e molto utili se si desidera conoscere molti parametri di default che non sono presenti nel main.cf a installazione completata. Ci si può inoltre aiutare usando il comando `postconf`. Le direttive vengono separate tra di loro e riunite in categorie (Queue and process ownership, Internet host and domain name, Sending mail...) facilitando in questo modo la comprensione del significato dei singoli parametri.

- ✓ `master.cf`: è il file che regola il comportamento del programma `master` responsabile della gestione degli altri programmi che compongono Postfix e i parametri con cui questi vengono eseguiti. Definisce, ad esempio, se un modulo deve girare in ambiente chroot e quante volte può essere avviato.
- ✓ `relocated`: in questo file sono presenti i nuovi indirizzi e-mail di utenti che si sono trasferiti. In questo modo ad esempio il messaggio `unknown user: "pippo"` viene sostituito da `user has moved to pippo@dominio.it`.
- ✓ `transport`: regola metodi speciali di trasporto per singoli domini. Qui vengono connessi, ad esempio, anche sistemi di scansione antivirus e sono inoltrate le informazioni raccolte da Postfix tramite DNS.
- ✓ `virtual`: consente di redirigere indirizzi di mittenti locali e virtuali a mailbox locali o ad altri indirizzi virtuali. Questo file trascrive indirizzi di posta solo in SMTP-Envelope, cioè, per `virtual` l'indirizzo del mittente originario rimane leggibile.

La directory radice dove si troveranno tutte le code è in `/var/spool/postfix/`. La documentazione verrà salvata in `/usr/share/doc` o in `/usr/doc` ma può cambiare a seconda della distribuzione usata.

La suite di postfix si compone di numerose applicazioni. Alla fine dell'installazione, se eseguita con i valori standard si troveranno tutti i demoni all'interno della directory `/usr/libexec/postfix/`.

```
[root@labreti4 ~]# ls -al /usr/libexec/postfix
totale 4160
drwxr-xr-x   2 root root   4096 25 mag 10:36 .
drwxr-xr-x  13 root root   4096 14 giu 12:41 ..
-rwxr-xr-x   1 root root 180020 11 feb  2005 bounce
-rwxr-xr-x   1 root root 219048 11 feb  2005 cleanup
-rwxr-xr-x   1 root root 167312 11 feb  2005 error
-rwxr-xr-x   1 root root 160112 11 feb  2005 flush
-rwxr-xr-x   1 root root 209648 11 feb  2005 lmtp
-rwxr-xr-x   1 root root 229328 11 feb  2005 local
-rwxr-xr-x   1 root root 103472 11 feb  2005 master
-rwxr-xr-x   2 root root 204804 11 feb  2005 nqmgr
-rwxr-xr-x   1 root root 200548 11 feb  2005 oqmgr
-rwxr-xr-x   1 root root 159304 11 feb  2005 pickup
-rwxr-xr-x   1 root root 188112 11 feb  2005 pipe
-rwxr-xr-x   1 root root 143636 11 feb  2005 proxymap
```

```
-rwxr-xr-x   2 root root 204804 11 feb  2005 qmgr
-rwxr-xr-x   1 root root 175716 11 feb  2005 qmqpd
-rwxr-xr-x   1 root root 163120 11 feb  2005 showq
-rwxr-xr-x   1 root root 256880 11 feb  2005 smtp
-rwxr-xr-x   1 root root 310660 11 feb  2005 smtpd
-rwxr-xr-x   1 root root 154940 11 feb  2005 spawn
-rwxr-xr-x   1 root root 180264 11 feb  2005 tlsmgr
-rwxr-xr-x   1 root root 167804 11 feb  2005 trivial-rewrite
-rwxr-xr-x   1 root root 154996 11 feb  2005 verify
-rwxr-xr-x   1 root root 184116 11 feb  2005 virtual
```

Le utility di gestione del servizio invece saranno all'interno della directory `/usr/sbin/`

```
[root@labreti4 ~]# ls -al /usr/sbin/post*
-rwxr-xr-x  1 root root     150612 11 feb  2005 /usr/sbin/postalias
-rwxr-xr-x  1 root root      78800 11 feb  2005 /usr/sbin/postcat
-rwxr-xr-x  1 root root     155764 11 feb  2005 /usr/sbin/postconf
-rwxr-sr-x  1 root postdrop  98696 11 feb  2005 /usr/sbin/postdrop
-rwxr-xr-x  1 root root      69840 11 feb  2005 /usr/sbin/postfix
-rwxr-xr-x  1 root root      77488 11 feb  2005 /usr/sbin/postkick
-rwxr-xr-x  1 root root      74032 11 feb  2005 /usr/sbin/postlock
-rwxr-xr-x  1 root root      71216 11 feb  2005 /usr/sbin/postlog
-rwxr-xr-x  1 root root     142420 11 feb  2005 /usr/sbin/postmap
-rwxr-sr-x  1 root postdrop 121780 11 feb  2005 /usr/sbin/postqueue
-rwxr-xr-x  1 root root      86992 11 feb  2005 /usr/sbin/postsuper
```

Passiamo ora ad una prima configurazione base per avere un sistema di posta funzionante. Prima di tutto è necessario convertire lo standard MTA in Postfix. Questa operazione si svolge semplicemente lanciando il comando:

```
# alternatives --config mta
```

Verrà presentato il seguente menù da cui si può scegliere Postfix digitando 2 e invio:

```
Ci sono due programmi che forniscono 'mta'.
         Selezione                                Comando
-----------------------------------------------------------------
*+           1                      /usr/sbin/sendmail.sendmail
             2                      /usr/sbin/sendmail.postfix
Premere invio per mantenere la selezione corrente [+], o inserire il numero di
selezione:
```

Di norma il `master.cf` può essere lasciato intatto a meno che non si stia lavorando su configurazioni più complesse. Occorre modificare poche variabili:

- Quale dominio usare per la posta in uscita;
- Quale invece per la posta in entrata;
- Quali client sono abilitati ad inviare la posta attraverso il server.

E' necessario quindi adattare alle proprie esigenze i parametri presenti nel file di configurazione `main.cf` presente nella cartella `/etc/postfix/`. Vediamo le direttive che ho modificato:

```
myorigin = $mydomain
```

è il dominio usato per la posta inviata dal server. Di default assume il valore della variabile `$myhostname` anche se è opportuno modificarlo con `$mydomain` soprattutto se non si utilizza il server per una piccola rete locale e se tale nome non è un FQDN (fully qualified domain name = nome completo di un host di Internet contrapposto al suo indirizzo numerico).

```
mydestination = $myhostname, localhost.$mydomain, $mydomain
```

specifica per quali domini il sistema di posta deve instradare localmente i messaggi anziché inviarli ad un altro server. Si tratta di una lista di nomi, si possono utilizzare anche file o tabelle. I primi due valori sono quelli di default. Di norma occorrerà aggiungere a questi `$mydomain` in modo che il server sia il mail server del dominio.

Postfix di default non permette l'invio di posta da parte di client e domini sconosciuti ovvero non è un open relay. Condizione necessaria per evitare che il proprio server venga utilizzato per l'invio di spam. Per consentire l'invio da parte dei client presenti all'interno della rete locale occorre abilitarli usando i seguenti parametri:

```
mynetworks = 192.168.0.0/24, 192.168.1.0/24, 127.0.0.0/8
```

Questo parametro rappresenta gli indirizzi IP dei client che sono autorizzati a spedire messaggi. In alternativa si può usare il parametro:

```
mynetworks_style = class
```

Impostando il valore `class` saranno considerati validi tutti i client appartenenti al network di classe A, B o C a cui il server appartiene. In alternativa è possibile lasciare il valore di default che è `subnet` che indica che Postfix accetterà di inviare la posta di tutti i client appartenenti alla stessa

sottorete locale del mail server oppure indicare il valore `host` tramite il quale l'invio sarà abilitato solo per la macchina locale.

```
myhostname = mail.labreti4.it
```

grazie a questo parametro è possibile impostare un valore che viene poi utilizzato nel file di configurazione attraverso la variabile `$myhostname`. Il valore di default usato da Postfix è il nome dell'host locale, tuttavia se tale valore non è nella forma FQDN oppure se non è un nome di dominio completo oppure ancora se fosse in ascolto su un'interfaccia virtuale occorre specificare il nome di dominio completo per il server (mail.dominio.it oppure mail.virtualdominio.it).

```
mydomain = labreti4.it
```

attraverso questo parametro si imposta il dominio a cui appartiene il server. Di default viene usato il valore di `$myhostname` togliendo la prima parte del nome (mail.dominio.it → dominio.it).
Nel caso in cui la macchina che si sta usando possiede più di un indirizzo su un'interfaccia di rete o più interfacce verrà utile specificare inoltre

```
inet_interfaces = all
```

determina su quali indirizzi vengono accettate le e-mail in arrivo. E' il valore di default, se si vuole impostare interfacce virtuali si imposta il valore `virtual.dominio.it`. Perchè la modifica di questa variabile sia effettiva non basta effettuare un reload ma si deve fermare e riavviare il sistema di posta.

14.1.3 Attivazione

Configurati questi parametri si dovrebbe cominciare ad avere un server di posta utilizzabile.
Il tipo di formato usato da Postfix è di default il mailbox quindi i messaggi di posta verrano consegnati nella classica cartella `/var/spool/mail` o `/var/mail`. Per verificare che postfix sia attivo e funzionante è necessario prima di tutto fermare Sendmail. Per far questo digitiamo il comando:

```
# /etc/rc.d/init.d/sendmail stop
```

A questo punto per dare il via al mail server basterà lanciare il comando

```
# postfix start
```

Se non vengono segnalati messaggi di errore il mail server è stato configurato con successo. In caso di problemi si può avviare il demone in modalità debug in modo da avere maggiori informazioni su quello che accade:

```
# postfix –D start
```

Per far rileggere le configurazioni senza riavviare il servizio è sufficiente digitare:

```
# postfix reload
```

Per far partire postfix direttamente al boot si può aggiungere al file `/etc/rc.d/rc.local` la seguente riga:

```
postfix start
```

Prima di utilizzarlo tuttavia è consigliato riavviare il server per essere sicuri che tutte le funzioni fondamentali vengano attivate. Verifichiamo quindi che Postfix sia attivo con i seguenti comandi:

```
[root@labreti4 ~]# ps –ax | grep postfix
3752 ?        Ss     0:00 /usr/libexec/postfix/master
3805 pts/2    S+     0:00 grep postfix

[root@labreti4 ~]# lsof -i :25
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
master  3752 root   11u  IPv4   8808       TCP *:smtp (LISTEN)

[root@labreti4 ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
quit
221 Bye
Connection closed by foreign host.
```

Creiamo due utenti (`amministratore` e `postino`) e inviamo quindi un'e-mail standard di prova con l'utility `mail` e con `telnet`:

```
[root@labreti4 ~]# mail postino
Subject: test da localhost

Test 1
.
Cc:

[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 3 messages 1 new 3 unread
 U  1 amministratore@labre  Fri Sep 30 17:14  16/550   "ciao"
 U  2 root@labreti4.labret  Fri Sep 30 17:19  22/710   "prova"
```

```
>N  3 root@labreti4.it      Fri Sep 30 17:50  19/653   "test da localhost"
& 3
Message 3:
From root@labreti4.it  Fri Sep 30 17:50:23 2005
X-Original-To: postino@labreti4.it
Delivered-To: postino@labreti4.it
Date: Fri, 30 Sep 2005 17:50:23 +0200
From: root <root@labreti4.it>
To: postino@labreti4.it
Subject: test da localhost


Test 1

& q
"/var/spool/mail/postino" complete

[root@labreti4 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
helo domain.com
250 mail.labreti4.it
helo labreti4.it
250 mail.labreti4.it
mail from: <test@domain.com>
250 Ok
rcpt to: <amministratore@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: test da localhost

test 2
.
250 Ok: queued as 69EE0C8D20
quit
221 Bye
Connection closed by foreign host.

[root@labreti4 ~]# mail -f /var/spool/mail/amministratore
```

```
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/amministratore": 6 messages 6 new
>N  1 akyra@interfree.it    Fri Sep 30 15:24  15/533   "e allora"
 N  2 pluto@paperino.it     Fri Sep 30 16:05  15/540   "ci provo"
 N  3 akyra@interfree.it    Fri Sep 30 16:57  15/545   "andiamo"
 N  4 postino@labreti4.it   Fri Sep 30 17:16  15/546   "kmail"
 N  5 qualcuno@qualcosa.it  Fri Sep 30 17:29  15/543   "ciao"
 N  6 test@domain.com       Fri Sep 30 17:56  15/534   "test da localhost"
& 6
Message 6:
From test@domain.com  Fri Sep 30 17:56:34 2005
X-Original-To: amministratore@labreti4.it
Delivered-To: amministratore@labreti4.it
Subject: test da localhost
Date: Fri, 30 Sep 2005 17:55:21 +0200 (CEST)
From: test@domain.com
To: undisclosed-recipients:;

test 2

& q
"/var/spool/mail/amministratore" complete
```

14.1.4 Comandi di amministrazione

E' giunto il momento di conoscere i principali comandi per l'amministrazione del mailserver, i quali riguardano la gestione del sistema e quindi possono essere impartiti solo dall'utente `root`.

Il principale e più importante è il comando `postfix` che serve a controllare il programma stesso. Con esso si avvia o si ferma il demone `master` del sistema di posta, si verifica lo stato o si ricarica la configurazione.

La sua sintassi è semplice e senza troppe opzioni, con vari sotto-comandi:

```
postfix [-Dv] [-c directory_file_conf] comando_interno
```

Opzioni:

`-D`: Usata solo con il sotto-comando `start` permette di lanciare i demoni di postfix sotto il controllo delle direttive di debug specificate nel `main.cf`.

`-v`: Anche questa opzione è orientata al debug del software.

`-c`: Permette di specificare una directory alternativa a quella di default in cui il comando andrà a cercare i files di configurazione. Utile nel caso si stiano eseguendo più server di posta sulla stessa

macchina.

Diamo una rapida occhiata anche ai comandi interni:

- ✓ `start`: Avvia il sistema di posta. Effettua anche un controllo della configurazione come se si usasse il comando `check`.
- ✓ `stop`: Ferma il sistema regolarmente. I processi in corso hanno modo di terminare quanto prima ma completando le loro operazioni.
- ✓ `abort`: Ferma il sistema forzatamente. I processi in corso sono costretti a fermarsi immediatamente, senza attendere che si concludano le operazioni in corso.
- ✓ `check`**:** Verifica la configurazione del sistema di posta, controlla eventuali permessi e proprietà errate e provvede a creare eventuali directory mancanti.
- ✓ `flush`: Forza l'inoltro di ogni messaggio presente nella coda `deferred` in attesa di essere inviato. Utile nel caso si usi una connessione verso Internet di tipo PPP o ISDN che non è sempre attiva.
- ✓ `reload`: Permette di ricaricare i files di configurazione senza fermare il sistema di posta.

Altri tool a corredo di Postfix e utili per l'amministrazione sono:

- ➢ `postalias`: permette di creare, interrogare e aggiornare il database degli alias. Quindi lanciando il comando:

  ```
  # postalias /etc/postfix/aliases
  ```

  creiamo o aggiorniamo il database degli alias. E' l'analogo del comando `newaliases` di Sendmail. Il file degli alias ha il seguente formato:

  ```
  nome: valore, valore, valore….
  ```

  nel quale `nome` indica un indirizzo locale e `valore` è un indirizzo standard o un file (dove è contenuta una lista di indirizzi) oppure un comando.
- ➢ `postcat`: elenca il contenuto delle code.
- ➢ `postconf`: permette di visualizzare e modificare i parametri di configurazione. Consente inoltre di modificare la configurazione di `main.cf` (ad esempio per gli script).
- ➢ `postdrop`: riceve le e-mail tramite linea di comando e le dispone nelle code.
- ➢ `postkick`: invia comandi di controllo ai singoli moduli Postfix.
- ➢ `postlock`: blocca le mailbox di Postfix. Ideale per gli script.
- ➢ `postlog`: crea messaggi di log per `syslogd`. Permette quindi di scrivere sul file di log un testo voluto da noi e ciò risulta molto utile nelle operazioni di debug. Se lanciato senza argomenti prende il testo dallo standard input.

- `postmap`: consente di creare e interrogare le tabelle di lookup di Postfix e quindi converte le tabelle in formato hash (*.db). Le tabelle presentano in ciascuna riga una coppia chiave valore e quindi eseguendo una ricerca su una chiave viene restituito il valore corrispondente. In esse vengono registrati i dati relativi agli alias, ai domini virtuali, agli indirizzi di posta elettronica degli utenti.
- `postqueue`: permette la gestione delle code di invio agli utenti. Tra le operazioni consentite c'è la possibilità di invio immediato e la visualizzazione dei messaggi in coda.
- `postsuper`: fa la manutenzione delle code e quindi le gestisce (cancellazione, reinserimento di messaggi in coda). Inoltre se lanciato senza argomenti o con l'argomento `-s -p` verifica ed eventualmente ripara incongruenze delle directory di Postfix.
- `mailq`: visualizza il contenuto delle code indicandone lo stato attuale o gli eventuali problemi di invio.
- `newaliases`: ha le stesse funzioni di `postalias`, è stato creato per compatibilità con Sendmail.
- `sendmail`: assicura la compatibilità con Sendmail. Le e-mail trasmesse vengono inoltrate a `postdrop`.

14.1.5 Configurazione antispam e controllo sulle connessioni

Per ottimizzare il servizio ed ottenere un maggior controllo delle sue funzioni è necessario definire altri parametri che ci permettono di controllare gli accessi, sia da un punto di vista di consumo delle risorse sia dal controllo di cosa può passare e cosa invece no per evitare, ad esempio, l'invio massiccio attraverso il server di spam.

Di default l'SMTP-server di Postfix accetta esclusivamente la posta per il network o il dominio locale e per i domini ospiti del server, in modo che dagli estranei non si accetti alcuna connessione. Esistono diversi modi per assumere maggior controllo su questa problematica usando liste di accesso in stile Sendmail o RBL (Real Time Block List). Per un approfondimento si può fare riferimento alla documentazione ufficiale disponibile sul sito di Postfix, www.postfix.org , o nella directory di sistema dedicata alla documentazione dei pacchetti installati. E' importante inoltre controllare le quantità di posta processabile, ad esempio per limitare il numero di processi contemporanei anche se si dispone di un limite massimo di 1000 client.

Vediamo i parametri più importanti che ho provveduto ad inserire nel file `main.cf`:

`default_process_limit = 100`

permette di controllare il numero di processi in uscita e in entrata. Di default è settato a 100 ma nel caso di una piccola rete casalinga andrebbe benissimo un valore di 10 mentre per un mail server principale di un provider andrebbe meglio un valore di 1000 o 10000.

Per controllare che il server non sia usato per un attacco tipo spam si usano alcune variabili che ai loro valori di default (che indico subito) dovrebbero andare benissimo:

`initial_destination_concurrency = 2`

il quale controlla quanti messaggi sono inizialmente inviati verso la stessa destinazione prima di attuare un invio multiplo.

`local_destination_concurrency_limit = 2`

che controlla quanti messaggi sono inviati contemporaneamente allo stesso recipiente locale.

`default_destination_concurrency_limit = 20`

che impone il limite di messaggi inviabili contemporaneamente alla stessa destinazione.

Ulteriori parametri permettono di limitare l'uso del server da parte di client sbadati o maliziosi, rallentando di fatto le loro connessioni fino a chiuderle. Il server incrementa un conteggio per sessione degli errori dovuti a richieste da parte dei client non riconosciute valide o sconosciute e di tutte le richieste che in qualche modo violano le regole. Questo conteggio si azzera quando un messaggio viene inviato correttamente. Se il conteggio aumenta il server limita i danni rallentando il client. Le variabili a cui ci riferiamo (impostate al loro valore di default) sono:

`smtpd_error_sleep_time = 1`

la quale cerca di evitare gli errori di un client quando il conteggio è ancora basso ponendo un tempo limite in cui il server resta in pausa mentre comunica l'errore al client.

`smtpd_soft_error_limit = 10`

stabilisce che quando il numero di errori per sessione supera il valore impostato il server attende il valore del conteggio in secondi prima di rispondere ad un'altra richiesta del client.

`smtpd_hard_error_limit = 20`

stabilisce che al superare nel conteggio del valore impostato il server SMTP taglia la connessione.

Vi sono poi numerosi parametri con cui è possibile effettuare diversi controlli sul traffico di posta del server. Presentiamo ora un elenco di ciò che è possibile fare:

1. *Stabilire il comportamento del server al momento della connessione del client*:

   I parametri per queste operazioni riguardano comandi interni al protocollo SMTP. Si può ad esempio costringere il server ad accettare connessioni strettamente aderenti allo standard stabilito nella RFC 821 o modificarne il comportamento in concomitanza con alcuni comandi SMTP.

   `# smtpd_helo_restrictions = valori`

   permette di definire diverse regole di comportamento del server in relazione a quali host possono inviare un comando HELO. Questo parametro io l'ho lasciato commentato.

   Oppure è possibile restringere i nomi dei client o i loro indirizzi IP con il parametro
   `smtpd_client_restrictions = hash:/etc/postfix/host-reject`

   oppure agire sul campo sender

   `smtpd_sender_restrictions = hash:/etc/postfix/address-reject`

   o sul comportamento del relay della posta (questo parametro l'ho lasciato commentato)
   `# smtpd_recipient_restrictions = valori`

   Per quanto riguarda i parametri relativi alle restrizioni sui client o sul mittente si possono specificare direttamente gli indirizzi o i nomi oppure (come ho fatto io) si possono indicare alcuni files di testo in cui verrà costruita la lista degli indesiderati indicando gli indirizzi e l'azione da intraprendere. Ad esempio per il file host-reject io ho indicato:

   ```
   192.168.0.2    REJECT
   ```

   E in address-reject

   ```
   esempio@reject.it   REJECT
   ```

   Per costruire le tabelle è necessario lanciare i comandi:

   ```
   # postmap /etc/postfix/host-reject
   # postmap /etc/postfix/address-reject
   ```

   Riporto ora un esempio riguardante il caso di mittente rifiutato dal file address-reject:

   ```
   root@labreti4 postfix]# telnet labreti4.it 25
   Trying 192.168.0.3...
   Connected to labreti4.it (192.168.0.3).
   Escape character is '^]'.
   220 mail.labreti4.it ESMTP Postfix
   mail from: <esempio@reject.it>
   ```

```
250 Ok
rcpt to: <amministratore@labreti4.it>
554 <esempio@reject.it>: Sender address rejected: Access denied
quit
221 Bye
Connection closed by foreign host.
```

Analizziamo anche il caso di host-reject:

```
root@labreti4 postfix]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
mail from: <prova@reject.it>
250 Ok
rcpt to: <amministratore@labreti4.it>
554 <unknown[192.168.0.2]>: Client host rejected: Access denied
quit
221 Bye
Connection closed by foreign host.
```

2. *Effettuare controlli sugli header di un messaggio e sul corpo dei messaggi:*

   Si possono stabilire regole di filtraggio sulle intestazioni di un messaggio contenenti informazioni sul tipo di programma client e sulle macchine attraverso cui il messaggio è passato. In questo modo si dà la possibilità di implementare delle liste nere di server da cui non accettare mai la posta e si possono anche filtrare i contenuti di un messaggio stabilendo liste di termini non permessi.

3. *Restringere l'inoltro della posta per quanto riguarda l'utente specificato nel campo "sender":*

   Si può stabilire che il server non accetti posta da client che non specificano un nome di dominio FDQN oppure accettare l'inoltro solo da utenti definiti. Il parametro che si occupa di questa operazione è

   `smtpd_sender_restrictions = hash:/etc/postfix/address-reject`

   La sua sintassi, simile nei parametri complementari visti in precedenza, permette di restringere l'uso del server di posta agendo sul campo sender. Opzioni possibili per questo sono：

   `reject_unknown_sender_domain`

   `reject_non_fqdn_sender`

oppure un file database con le specifiche per i sender autorizzati con l'utilizzo di una sintassi particolare sintassi *tipo_file:percorso_file* come ad esempio

```
hash = /etc/postfix/access
```

4. *Agire sui recipient di inoltro della posta:*

   Stabilendo regole sulla funzione di relay del server e agendo in sostanza sugli indirizzi concessi nel campo SMTP `RCPT TO` in modo da limitare o dare accesso per il relay della posta a client non appartenenti alla sottorete del server. Questo metodo è consigliato quando si utilizzano controlli di accesso tipo "pop-before-smtp" che permette l'uso del relay solo a client che si sono autenticati via pop precedentemente, altrimenti potrebbe essere usato come open relay.

   Il parametro principale è

   ```
   smtpd_recipient_restrictions = valori
   ```

   le cui opzioni impongono che sia presente sempre una voce tipo `reject`, `defer`, `defer_if_permit` o `reject_unauth_destination` altrimenti il server rifiuterà comunque la ricezione di posta.

   Per esempio io ho impostato:

   ```
   smtpd_recipient_restrictions = permit_mynetworks,
                                  reject_unauth_destination
   ```

   Questo parametro è utilizzabile anche specificando un'opzione di controllo sui client `check_client_access = tipo_file:percorso_file`

14.1.6 Alias dei domini locali

Per avere la possibilità di usare gli alias è necessario inserire nel file `main.cf` la seguente riga:

```
alias_maps = hash:/etc/postfix/aliases
```

Gli alias di posta locali redirigono le e-mail destinate ad un utente locale, come per esempio le e-mail di sistema del nostro server a un utente.

Per aggiungere o modificare alias è necessario editare il file `aliases`.

E' un file di testo, può contenere linee vuote e di commento (le quali iniziano con il carattere #). Può risiedere in `/etc/posftix/` oppure in `/etc/`.Ogni riga deve essere nel formato `nome:valore`.

La parte `nome` di ogni linea di configurazione è intesa come locale, di conseguenza non è necessario aggiungere il nome di dominio.

Per quanto riguarda la parte `valore`, essa può essere può contenere diversi tipi di destinazione:

- `address`: Un indirizzo e-mail;
- `file/name`: La mail viene accodata al file specificato. E' possibile utilizzare normali file, ma anche file speciali come `/dev/null`;
- `| comando`: Le mail possono essere inviate tramite pipe ad un comando;
- `:include:/file/name`: La mail viene inviata agli indirizzi elencati nel file specificato;

Nel file `etc/postfix/aliases` sotto il commento `#Person who should get root's mail` ho inserito quindi la seguente riga per indicare che le e-mail dirette all'utente `root@labreti4.it` sono reindirizzate all'utente `amministratore@labreti4.it`.

```
# Person who should get root's mail
root:      amministratore
```

Una volta aggiunti i nuovi alias, al fine di rendere effettive le modifiche, è necessario lanciare il comando `newaliases`, il quale si occuperà di aggiornare il database degli alias oppure `postalias /etc/postfix/aliases`.

14.1.7 Domini di posta virtuali

Per poter usufruire degli indirizzi virtuali è necessario inserire la seguente linea nel file `main.cf`:

```
virtual_maps = hash:/etc/postfix/virtual
```

Nel file `/etc/postfix/virtual` si può indicare a Postfix dove trovare la lista degli indirizzi che possono essere accettati e a quale utente locale consegnare i messaggi. Si tratta di una redirezione degli indirizzi che può essere fatta su indirizzi locali e non locali e su interi domini.

Nel file `/etc/postfix/virtual` ho quindi inserite le seguenti direttive:

```
labreti4.it              anything
postmaster@labreti4.it   admin
postino@labreti4.it      postino
spam@labreti4.it          spam
@labreti4.it              amministratore@labreti4.it
```

La prima riga stabilisce che tutte le e-mail del dominio, non assegnate ad alcuna casella vengono rimbalzate. La seconda accerta che la casella postmaster sia stata creata. L'ultima indica che tutte le e-mail indirizzate al dominio `labreti4.it` andranno reindirizzate all'indirizzo `amministratore@labreti4.it` quindi tutte le e-mail che non sono state definite sono indirizzate ad amministratore.

Per realizzare la tabella di lookup dovremo lanciare il comando:

```
# postmap /etc/postfix/virtual
```

14.1.8 Raccolta di e-mail dal server (POP/IMAP): Dovecot

Per richiamare le email tramite POP3/IMAP4 ho installato il pacchetto `dovecot` (http://www.dovecot.org). Dovecot è un server IMAP e POP3 opensource per sistemi LINUX/UNIX-like con supporto per i protocolli sicuri IMAPS e POPS, TLS e SSL. Può lavorare con caselle di posta in entrambi i formati: mbox e Maildir. E' facilmente configurabile, richiede l'autenticazione degli utenti per la ricezione della posta anche per mezzo di sistemi di autenticazione come PAM. Infatti prima di concedere l'accesso va a verificare se esiste l'utente e soprattutto se la password è corretta nel file `/etc/passwd`.

Per quanto riguarda il mio caso ho adattato il file di configurazione `/etc/dovecot.conf` cambiando pochi dettagli in quanto le impostazioni di default di Dovecot sono perfettamente ragionevoli. Più precisamente ho modificato i seguenti parametri:

- `protocols` inserendo i protocolli a cui ero interessata, ovvero IMAP e POP3 (`protocols = imap pop3`);
- `base_dir` segnalando la directory dove memorizzare i dati runtime;
- `default_mail_env` indicando la cartella nella quale dovecot potrà trovare le mailbox degli utenti (`default_mail_environment = mbox:/var/mail/%u`);
- `first_valid_uid = 500, last_valid_uid=0, first_valid_gid=1, last_valid_gid` settandoli ai loro valori di default. Questi parametri indicano il range degli utenti autorizzati.
- `auth_userdb=passwd` e `authpassdb=pam` indicando il metodo per verificare gli utenti e le loro password.

Ho fatto partire poi il servizio con il comando

```
# service dovecot start
```

Con il seguente comando ho impostato che si avviasse al boot

```
# chkconfig dovecot on
```

e infine ho verificato che i servizi erano attivi (prima SMTP che era già attivo poi POP e infine IMAP):

```
[root@labreti4 ~]# lsof -i :25
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
master  2153 root   11u  IPv4   5635       TCP *:smtp (LISTEN)
[root@labreti4 ~]# lsof -i :110
COMMAND    PID    USER   FD   TYPE DEVICE SIZE NODE NAME
dovecot   2090    root    6u  IPv4   5437       TCP *:pop3 (LISTEN)
```

```
pop3-logi 2136 dovecot    0u  IPv4   5437       TCP *:pop3 (LISTEN)
pop3-logi 2137 dovecot    0u  IPv4   5437       TCP *:pop3 (LISTEN)
pop3-logi 2138 dovecot    0u  IPv4   5437       TCP *:pop3 (LISTEN)
[root@labreti4 ~]# lsof -i :143
COMMAND    PID    USER   FD   TYPE DEVICE SIZE NODE NAME
dovecot   2090    root    5u  IPv4   5436       TCP *:imap (LISTEN)
imap-logi 2133 dovecot    0u  IPv4   5436       TCP *:imap (LISTEN)
imap-logi 2134 dovecot    0u  IPv4   5436       TCP *:imap (LISTEN)
imap-logi 2135 dovecot    0u  IPv4   5436       TCP *:imap (LISTEN)
```

14.2 SOFTWARE ANTISPAM: SPAMASSASSIN

14.2.1 Installazione

Come già detto, non ho dovuto provvedere all'installazione di Postfix, tuttavia per chiarezza spiegherò brevemente come avviene.

Il più delle volte è sufficiente installare i seguenti pacchetti (tenendo conto delle dipendenze):

```
rpm –i perl-Digest-SHA*
rpm –i perl-HTML-Tagset*
rpm –i perl-HTML-Parser*
rpm –i perl-Net-DNS*
rpm –i perl-Time-HiRes*
rpm –i spamassassin*
```

Si possono scaricare direttamente i pacchetti dal sito www.spamassassin.org.

Il pacchetto SpamAssassin, al suo interno contiene:

- Il tool da riga di comando `spamassassin`;
- Il modulo `Mail::Spamassassin` che funziona da plugin per il modulo `Mail::Audit` e permette di utilizzare facilmente le funzioni di filtraggio antispam all'interno dei propri script Perl;
- Il demone in Perl `spamd`, che svolge le stesse funzioni del tool `spamassassin`;
- Il client `spamc`, scritto in C, il quale permette, in collaborazione con `spamd`, di gestire grossi volumi di traffico senza dover eseguire un interprete Perl per ogni singolo messaggio;
- `sa-learn`, il tool che si occupa di allenare i filtri bayesiani.

Nella cartella `/usr/share/spamassassin/` si trova la banca dati centrale con le impostazioni standard. Questi test vengono costantemente aggiornati di versione in versione e hanno tipicamente la seguente struttura:

```
zona IDENTIFICATIVO controllo
describe IDENTIFICATIVO Descrizione
```

`score IDENTIFICATIVO punteggio`

dove `zona` può essere `header`, `body`, `uri`, a seconda che la regola debba essere applicata all'intestazione del messaggio, al contenuto oppure a tutti gli indirizzi web trovati nel messaggio. Dando il comando:

`# grep ^uri /usr/share/spamassassin/*.cf` si possono quindi vedere gli URI che sono stati messi in blacklist.

I controlli sul contenuto invece possono essere semplici regexp (regular expression), ma anche chiamate a funzioni del modulo `Mail::SpamAssassin` ad esempio per i controlli sulle liste RBL. `IDENTIFICATIVO` è il nome (scritto in maiuscolo) con il quale ci si riferisce a quel particolare filtro nel file di configurazione. `describe` associa a ciascuna regola una breve descrizione mentre `score` indica il punteggio da aggiungere o togliere al totale se il test ha successo.

Le regole sono suddivise in base alla `zona` a cui si riferiscono quindi ad esempio i tests sulle intestazioni sono in `20_head_tests.cf`, quelli sugli uri in `20_uri_tests.cf`, mentre i punteggi si trovano tutti in `50_scores.cf`. Esaminando quest'ultimo file si può notare che è possibile specificare un unico punteggio oppure quattro distinti. Questo perché se ne viene specificato uno solo, verrà utilizzato sempre quel valore, al contrario se ne sono specificati quattro ne verrà scelto uno a seconda che i filtri bayesiani e RBL sono abilitati o meno, seguendo questa tabella (dove il – indica disabilitato e il + indica abilitato):

`-Bayes -RBL: 1° valore`

`-Bayes +RBL: 2° valore`

`+Bayes -RBL: 3° valore`

`+Bayes +RBL: 4° valore`

Per personalizzare il funzionamento del filtro è possibile utilizzare il file di configurazione `local.cf` presente nella cartella `/etc/mail/spamassassin/`, il quale viene letto sempre per ultimo e quindi le opzioni in esso contenute hanno la meglio su tutte le altre.

Prima di passare alla configurazione vediamo altri files usati da SpamAssassin:

- ✓ `/usr/share/spamassassin/user_prefs.template`: questo file campione definisce le impostazioni specifiche dell'utente per SpamAssassin. Al momento dell'upgrade viene sovrascritto.
- ✓ `/etc/mail/spamassassin/user_prefs.template`: questo file campione può essere creato dall'amministratore e ha la priorità sul file campione in dotazione.
- ✓ `~user/.spamassassin`: questa directory contiene i files specifici dell'utente.

- ✓ `~user/.spamassassin/user_prefs`: qui l'utente può configurare spamassassin in base alle proprie esigenze. Se questo file non esiste, la prima volta che viene lanciato SpamAssassin uno dei files campione viene copiato in questo punto.

14.2.2 Integrazione con Postfix: Procmail

Prima di configurarlo è necessario collegare SpamAssassin al momento dell'inoltro delle e-mail. Il metodo più semplice per integrare SpamAssassin e Postfix è attraverso Procmail, che in Fedora è già installato. Apriamo una parentesi su Procmail.

Procmail è un Mail Delivery Agent (MDA), ovvero è utilizzato dall'MTA per consegnare le e-mail a una mailbox specifica di un utente. In particolare si tratta di un Local Delivery Agent (LDA). Questi programmi gestiscono i messaggi da consegnare fino al momento in cui possono essere letti da un MUA, infatti distribuisce messaggi sulla macchina locale per far accedere un'applicazione client. Gli stessi MTA a volte potrebbero funzionare come MDA in particolare quando aggiungono nuovi messaggi e-mail ad un file spool di posta dell'utente locale.

Procmail ha quindi il compito di consegnare le e-mail ovvero di spostarle dal file di spool dell'MTA alla mailbox dell'utente ed inoltre offre un sistema di filtraggio. E' ampiamente utilizzato, potente e non intrusivo. Può essere chiamato in diversi modi. Innanzitutto viene lanciato ogni volta che un MTA posiziona un'e-mail nel file di spool, svolge il suo compito di filtrare e archiviare per il MUA e poi abbandona l'applicazione. Oppure si può configurare il MUA in modo tale che ogni volta che riceve un messaggio faccia partire Procmail per archiviare le e-mail nelle corrette mailbox. Per default viene utilizzata la prima opzione. Il file di configurazione è `/etc/procmailrc` oppure `.procmailrc` nella home directory dell'utente. Le condizioni e i requisiti di filtraggio vengono appunto definite in questo file ed in base a questo verranno quindi decise le sorti dell'e-mail che può essere spostata in un file specifico, cancellata o processata. Procmail legge il messaggio, separa la parte dell'informazione che gli interessa, poi va alla ricerca del file di configurazione per confrontarlo con le regole e le variabili d'ambiente che indicano i messaggi da smistare anche verso caselle di posta alternative e/o altri indirizzi di posta e come comportarsi nei confronti dei messaggi che non soddisfano alcuna regola. Sono gli stessi utenti che definiscono le regole e creano il file `.procmailrc` nella propria home directory in base alle loro esigenze.

Innanzitutto è necessario inserire la seguente riga nel file main.cf di Postfix, in modo che al momento dell'inoltro Postfix lanci Procmail:

```
mailbox_command  = /usr/bin/procmail
```

Affinchè Procmail lanci solo SpamAssassin, ogni utente deve disporre di un file `.procmailrc` nella propria home directory. Se gli utenti non possiedono file `.procmailrc` personali, può

essere utilizzato anche il file di sistema `/etc/procmailrc`, che ho provveduto a creare, valido poi per tutti gli utenti. Riporto ora le modiche da me apportate:

```
DROPPRIVS=yes
LOGFILE=/var/log/procmail.log
:0fw: spamassassin.lock
* < 256000
| /usr/bin/spamassassin

:0:/tmp/spam.lock
*^X-Spam-Flag: YES
/var/spool/mail/spam
```

Come si può notare è necessario inserire la riga:

```
DROPPRIVS=yes
:0fw: spamassassin.lock
| /usr/bin/spamassassin
```

In questo modo, prima di essere inoltrata, ogni email viene filtrata da SpamAssassin (viene specificato infatti di usare il binario `/usr/bin/spamassassin`). Per sistemi che ricevono molte e-mail tuttavia questo metodo richiede un carico notevole poiché per ogni e-mail viene lanciato un processo di SpamAssassin. In questo caso è consigliato settare la versione demone di SpamAssassin inserendo queste righe:

```
:0fw: spamassassin.lock
| /usr/bin/spamc
```

Con `LOGFILE` specifichiamo il file dove procmail logga le attività che svolge e con `* < 256000` Specifichiamo a procmail di chiamare spamassassin solo per le email più piccole di 256K, in caso contrario si rischia di appesantire il processo di parsing delle email stesse. Infine con la direttiva

```
-*^X-Spam-Flag: YES
/var/spool/mail/spam
```

se trova il flag nell'header dell'e-mail (inserito da spamassassin per le email ritenute spam) sposta la mail in una mailbox chiamata appunto "spam".

Per avviare il servizio lanciare il comando:

```
# service spamassassin start
```

e per fare in modo che ciò avvenga in modo automatico all'avvio del sistema è necessario lanciare quest'altro comando:

```
# chkconfig spamassassin on
```

A questo punto SpamAssassin è già in grado di analizzare le e-mail e di assegnarle le corrispondenti intestazioni.

14.2.3 Configurazione

La configurazione di SpamAssassin può avvenire in diversi files. A livello di sistema si modifica il file `/etc/mail/spamassassin/local.cf`. L'utente può sovrascrivere queste impostazioni nel suo file personale `~user/.spamassassin/user_prefs`.

Analizziamo ora le regole principali che si possono impostare.

Il parametro `required_hits` rappresenta il valore limite per considerare un'e-mail spam. Se si abbassasse il valore l'impostazione sarebbe più sensibile e più e-mail verrebbero classificate come spam con il rischio però di aumentare i falsi positivi.

E' quindi più conveniente modificare il punteggio di singoli tests attraverso il parametro `score` aumentando o diminuendo il punteggio assegnato a determinate regole.

Spamassassin aggiunge intestazioni alle e-mail ma può anche riscrivere l'oggetto delle e-mail per rendere più evidente l'individuazione dello spam, attraverso i parametri `rewrite_subject 1` con la stringa `subject_tag`. Per impedire il cambiamento impostare `rewrite_subject` a 0.

Il comportamento delle intestazioni può essere modificato attraverso i parametri `add_header`, `remove_header` e `clear_headers`.

Anche il corpo del messaggio viene modificato, aggiungendo i risultati del test, che possono essere inseriti in forma più concisa con `use_terse_report 1` oppure possono essere messi nell'intestazione con `report_header 1`.

Si possono impostare anche le lingue utilizzando i parametri `ok_languages` e `ok_locales`. In questo modo le e-mail in altre lingue vengono più probabilmente riconosciute come spam.

Inoltre per riconoscere le e-mail in base agli indirizzi si utilizzano i seguenti parametri:

`whitelist_from` e `blacklist_from` in grado di filtrare il messaggio in base all'indirizzo del mittente. Per rimuovere indirizzi specifici dalle liste si usano `unwhitelist_from` e `unblacklist_from`. Può essere utile abbinare le opzioni `whitelist_from` e `unwhitelist_from` se si vogliono leggere tutte le e-mail provenienti da un dominio senza che vengano filtrate tranne quelle di un certo utente di cui non ci si fida molto e su cui, quindi, si vogliono effettuare controlli. Con le stesse opzioni riguardanti però le blacklist, si può fare il contrario, cioè fidarsi solo di alcuni utenti in un dominio screditato.

Per filtrare l'indirizzo del destinatario SpamAssassin offre i parametri `whitelist_to` e `blacklist_to`, comodi soprattutto per le mailing list che possono venire bersagliate da spam. Oltre a `whitelist_to` esistono anche `more_spam_to` e `all_spam_to` che permettono di aggiungere un indirizzo alla whitelist ma a livelli diversi. Il primo esegue controlli in modo poco severo, il secondo chiude un occhio sulla maggior parte dei test e il terzo non filtra per niente. Non esiste il corrispondente `unwhitelist_to`. E' possibile infine accettare un'e-mail anche in base all'indirizzo IP di invio tramite il parametro `trusted_networks`. In tutti questi casi non vengono valutate le liste nere, si dà precedenza a queste regole.

Con i parametri `header` e `body` si possono definire regole anche con espressioni regolari.

Con la direttiva `report safe 1` si stabilisce che se un messaggio è targato come spam invece di modificare il messaggio originale SpamAssassin creerà un nuovo messaggio di report.

Con la riga `allow_user_rules 1` si dà la possibilità a tutti gli utenti di creare regole personali nel proprio file personale.

Infine è anche possibile utilizzare banche dati esterne. In particolare SpamAssassin può ricorrere a RBL, Razor 2 di Vipul, DCC o Pyzor. Come impostazione predefinita esiste solo l'utilizzo delle RBL. Non ho provveduto a scaricare i pacchetti di Razor2, DCC e Pyzor in quanto lavorando localmente non ne esisteva la necessità.

Nel caso si disponesse di una connessione lenta e si volesse saltare il controllo sulle blacklist che aumenta il carico di rete si deve inserire la seguente riga: `skip_rbl_checks 1`. Inoltre si può usare `check_mx_delay` che abbassa il tempo massimo di attesa per verificare che l'indirizzo di provenienza sia plausibile (di default è 5 secondi).

Se ancora non bastasse si può disattivare completamente questo controllo con `check_mx_attemps 0`.

La descrizione dettagliata di tutte le altre opzioni è disponibile eseguendo da terminale il comando

```
# man Mail::SpamAssassin::Conf
```

Riporto ora le modifiche da me apportate per configurare il file `local.cf`:

```
# Imposto la soglia superata la quale si considera il messaggio come spam
required_hits 5

# Riscrivo l'oggetto con la stringa SPAM
rewrite_header Subject [SPAM]

# Imposto le lingue da cui l'utente si aspetta le email
ok_languages it en
ok_local it

# Permetto agli utenti di definire proprie regole
allow_user_rules 1
```

```
# Aumento il punteggio assegnato per la parola TONER
score TONER     2.5

# Aumenta il punteggio assegnato per lingue non desiderate
score UNDESIRED_LANGUAGE BODY   2

#Aumento il punteggio dei messaggi contenenti Undisclosed-recipients
score UNDISC_RECIPS     1.5

# Aumento il punteggio dei messaggi formattati in HTML
score HTML_MESSAGE      1.5

# Inserisco una regola per messaggi il cui subject inizia con [OT]
header ANTI_TROLL Subject =~ /^\[OT\]/
describe ANTI_TROLL Subject: found OT tag
score ANTI_TROLL        3.5

# Inserisco in whitelist tutti gli indirizzi appartenenti al dominio labreti4.it
whitelist_from  *@labreti4.it

# Inserisco in blacklist l'indirizzo pluto@paperino.it
blacklist_from  pluto@paperino.it

# Aumento il punteggio per la parola VIAGRA
body VIAGRA     /viagra/i
score VIAGRA    7

# Imposto il cambiamento del messaggio se viene targato come spam
report safe     1

# Disabilito il salto del controllo sulle RBL
skip_rbl_checks 0

# Imposto a 10 il limite di tempo massimo per effettuare il controllo sulle RBL
rbl_timeout 10

# Abilito il controllo con i filtri bayesiani
use_bayes       1
use_bayes_rules 1

# Disabilito l'auto learning
bayes_auto_learn        0

# Imposto la variabile per il database dei filtri bayesiani
bayes_path /var/spool/spamassassin/bayes
```

14.2.4 Allenare i filtri bayesiani

SpamAssassin è uno dei pochi tool antispam che offre il supporto ai filtri bayesiani. Per far interpretare il contenuto del messaggio al filtro e anche per ottenere i migliori risultati possibili è necessario allenare il più a lungo possibile il software, fornendogli una certa quantità di spam ed un elevato numero di messaggi buoni. Il comando mediante il quale è possibile realizzare questo allenamento è `sa-learn`. Occorrerà quindi recuperare e utilizzare per creare un database per il filtro due mailbox: una in formato mbox o Maildir, nella quale siano stati salvati solo messaggi di

spam e una nella quale siano stati salvati solo messaggi buoni. Prima di procedere si deve specificare in `local.cf` il percorso del database globale con la seguente regola:

```
bayes_path /var/spool/spamassassin/bayes
```

Inoltre è necessario specificare i seguenti parametri per abilitare l'uso dei filtri bayesiani:

```
use_bayes 1
use_bayes_rules 1
```

Postfix per default usa il formato mbox quindi io ho dovuto usare i seguenti comandi:

```
sa-learn --mbox --spam bad.mbox
sa-learn --mbox --ham good.mbox
```

altrimenti per il formato Maildir si deve utilizzare:

```
sa-learn --spam /path/Maildir-bad/
sa-learn --ham /path/Maildir-good/
```

In particolare per i messaggi indesiderati ho usato la cartella spam e per quelli buoni la cartella postino. Vediamo ora il contenuto delle cartelle e l'esecuzione del comando sa-learn.

```
[root@labreti4 ~]# mail -f /var/spool/mail/spam
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/spam": 3 messages 2 unread
    1 spam@labreti4.it      Mon Oct 10 12:43  72/2477  "[SPAM] VIAGRA"
>U  2 postino@labreti4.it   Wed Oct 12 14:38  87/2817  "Fwd: [SPAM] ciao"
 U  3 postino@labreti4.it   Wed Oct 12 14:38  87/2817  "Fwd: [SPAM] ciao"
& q
[root@labreti4 ~]# sa-learn --mbox --spam /var/spool/mail/spam
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Learned from 3 message(s) (3 message(s) examined).
[root@labreti4 ~]#

[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 3 messages 2 unread
>U  1 amministratore@labre  Wed Oct 12 14:44  27/987   "ciao"
 U  2 root@labreti4.it      Wed Oct 12 14:44  19/655   "test"
```

```
    3 root@labreti4.it      Wed Oct 12 14:44  19/658   "prova"
& q
[root@labreti4 ~]# sa-learn --mbox --ham /var/spool/mail/postino
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Argument "BODY" isn't numeric in addition (+) at
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin/Conf.pm line 244.
Learned from 3 message(s) (3 message(s) examined).
```

Via via che si riceve nuova posta e nuovo spam è bene ripetere la procedura. `sa-laern` leggerà solo i messaggi nuovi che sono stati aggiunti nella mailbox. Se per errore si sbaglia a mettere i messaggi nelle cartelle corrette basterà rimettere tutto a posto e rilanciare il comando.

Ogni mail server si trova a gestire una tipologia di messaggi con caratteristiche diverse dagli altri e lo scopo dei filtri bayesiani è proprio quello di identificare queste caratteristiche. Quindi non servirebbe usare un database globale valido per tutti i mail server.

L'utente deve raccogliere una gran quantità (almeno 100) di e-mail SPAM e NOSPAM (HAM). Non appena SpamAssassin ha imparato abbastanza e-mail, le regole diventano attive automaticamente. Dopo il primo avvio, le regole rimangono inattive finchè la banca dati BAYES non contiene abbastanza record di dati.

Inoltre è possibile automatizzare la procedura di learning dello spam creando un account adatto allo scopo. In questo modo ogni volta che un utente riceverà spam potrà inoltrarlo a questo account. L'indirizzo che ho usato per questo scopo è `spam@labreti4.it`. E' sufficiente poi dare l'accesso in scrittura al database specificato tramite la variabile `bayes_path` e aggiungere le seguenti righe al file `.procmailrc` dell'utente `spam`:

```
LOGFILE=$HOME/procmail.log
:0
        {
        :0c: .spamassassin.spamlock
        | /usr/bin/sa-learn --spam

        :0: .spamassassin.filelock
        spam.mbox
   }
```

SpamAssassin può imparare anche in automatico le caratteristiche specifiche delle e-mail SPAM e NOSPAM e le aggiunge nella banca dati BAYES (autolearning). La direttiva cui far riferimento è `bayes_auto_learn`, la quale settata a 0 è disabilitata. In pratica se abilitata viene eseguita l'operazione di learning in modo automatico sui messaggi il cui `score` è estremamente elevato, in positivo o negativo come specificato nelle seguenti opzioni in `10_misc.cf`:

```
bayes_auto_learn_threshold_spam 12.0
bayes_auto_learn_threshold_nonspam 0.1
```

Per ottenere un punteggio elevato è necessario che il messaggio non superi molti test, e questo succede tipicamente quando un messaggio è proprio indesiderato.

Tuttavia questa opportunità spesso può rivelarsi un tallone d'Achille in quanto gli spammer hanno iniziato ad utilizzare una tecnica particolare detta Bayesian poisoning, che consiste nel sommergere i mail server con messaggi opportunamente redatti per ingannare il filtro bayesiano ed allenarlo a riconoscere i messaggi di spam come buoni. Quindi è consigliato lasciare disabilitata questa funzione ed allenare personalmente i filtri.

14.3 CLIENT DI POSTA: KMAIL

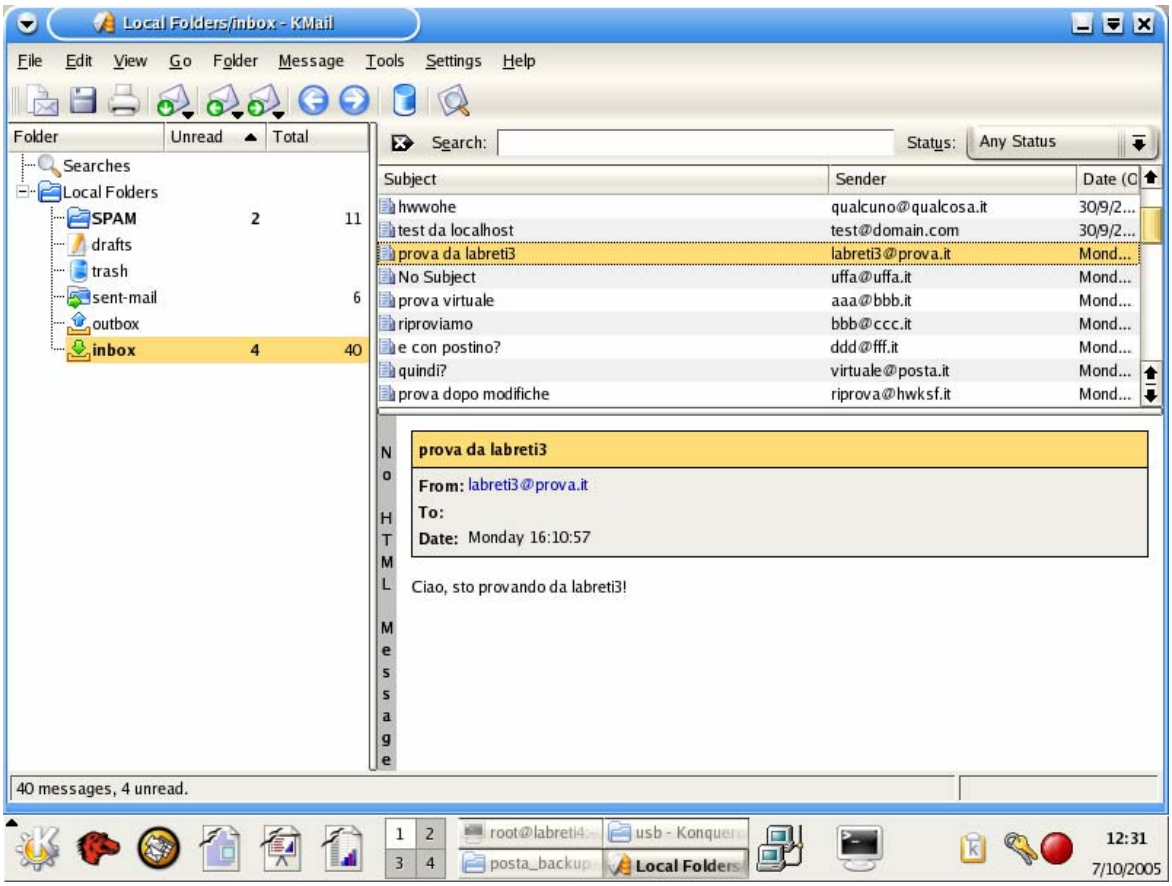
14.3.1 Configurazione

SpamAssassin si limita a visionare una per una le e-mail che gli vengono sottoposte, per poi apporre il suo timbro. Il resto è compito di altri programmi. E' l'utente a decidere cosa fare del messaggio, configurando regole opportune nel proprio client di posta. Nel mio caso la posta viene scaricata con Postfix, poi viene passata a Procmail che la controlla con SpamAssassin e la smista ed infine viene letta con un client di posta, nel quale vengono configurate alcune regole opportune per l'integrazione con SpamAssassin. Non tutti i client supportano questo tipo di filtraggio tramite programmi esterni, per questo ho utilizzato KMail che invece possiede questa opzione.

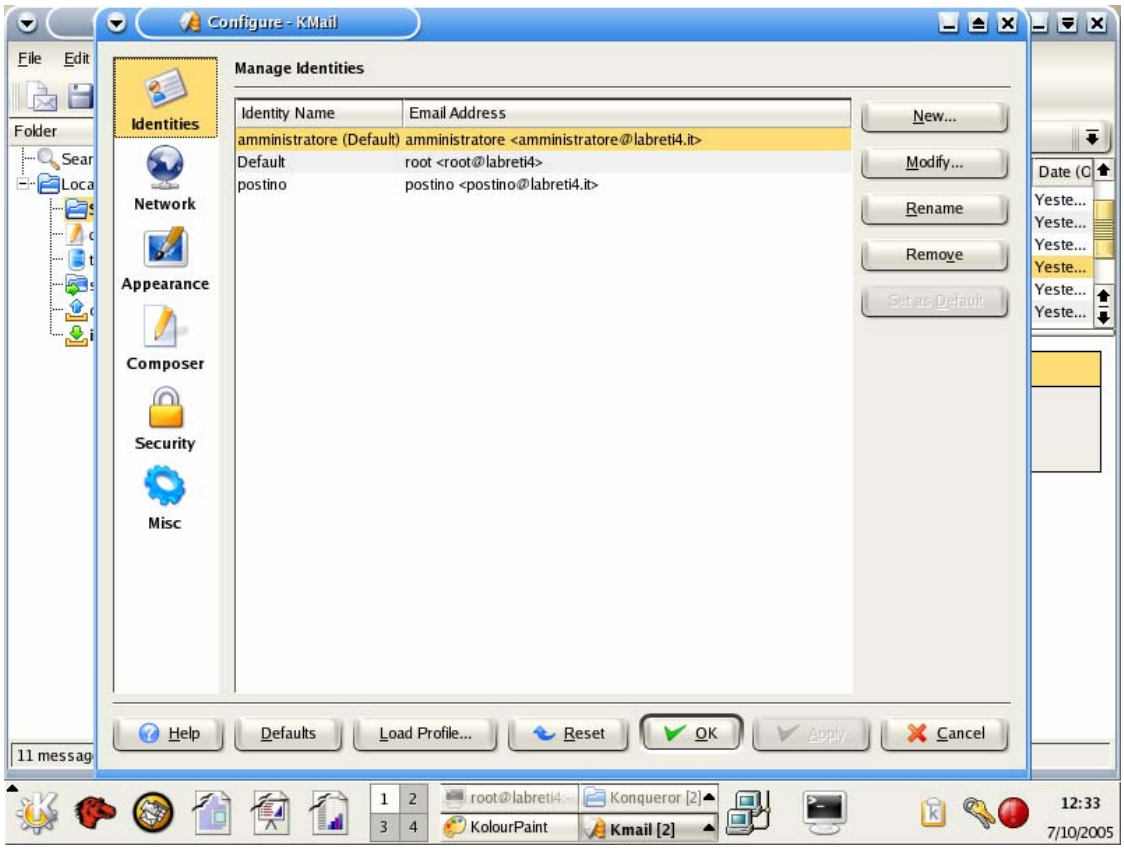
Per integrare SpamAssassin con KMail è necessario utilizzare i filtri in modo da passare la mail al programma ed agire in base al suo responso. Prima di tutto va creata un cartella destinata ad ospitare i messaggi indesiderati. Andranno a questo punto configurati i filtri. Si seleziona quindi la voce "Configures filters" dal menù "Setting". Si crea un nuovo filtro con il pulsante in basso a sinistra e si procede impostando i criteri. Nella mia configurazione ho impostato il controllo del campo "X-Spam-Flag" in modo che sia uguale a YES e il controllo del contenuto del messaggio in modo che sia contenuta la stringa [SPAM]. Come azione ho stabilito di spostare i messaggi nella cartella SPAM e l'inoltro del messaggio all'indirizzo `spam@labreti4.it` che utilizzo per allenare i filtri bayesiani.

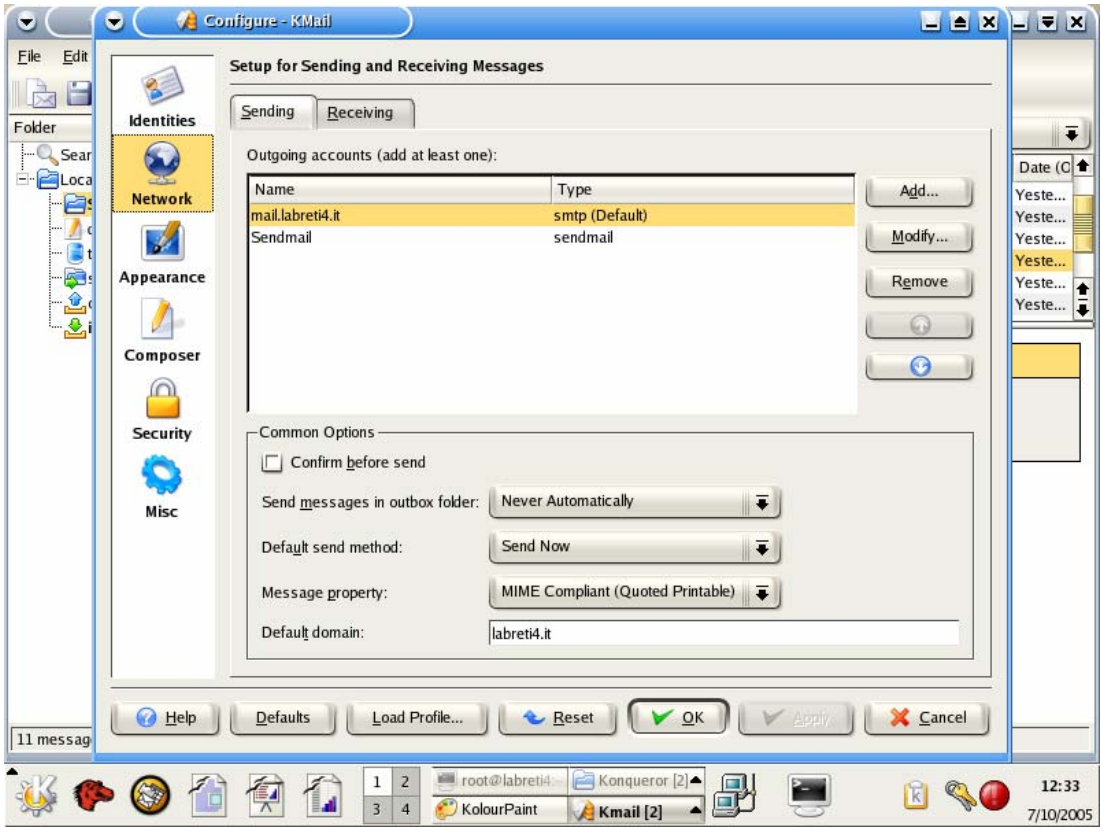
Per quanto riguarda poi le altre impostazioni sotto il menù “Settings” alla voce “Configure KMail” ho inserito gli account `amministratore` e `postino` oltre al predefinito `root` e ho configurato la spedizione con il protocollo SMTP e la ricezione con il protocollo POP.

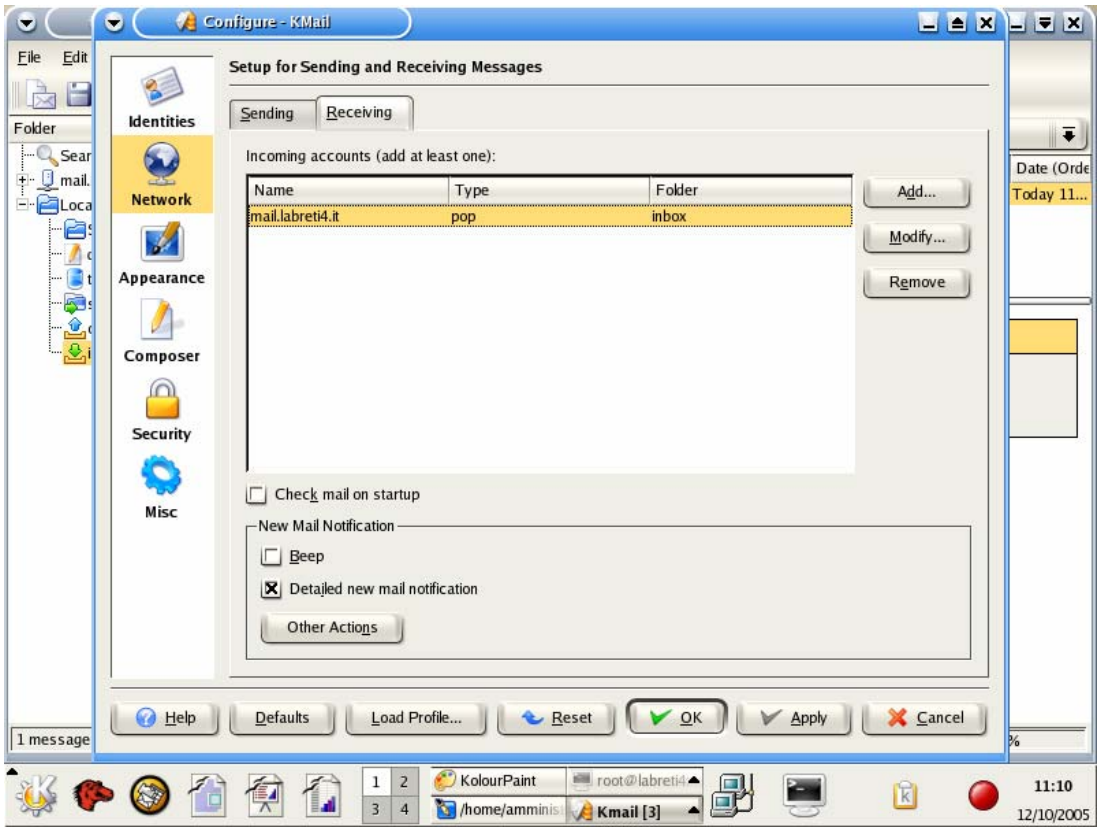
Riporto ora alcune figure che permettono meglio di comprendere questa configurazione. Nella prima si può notare la schermata principale di KMail:



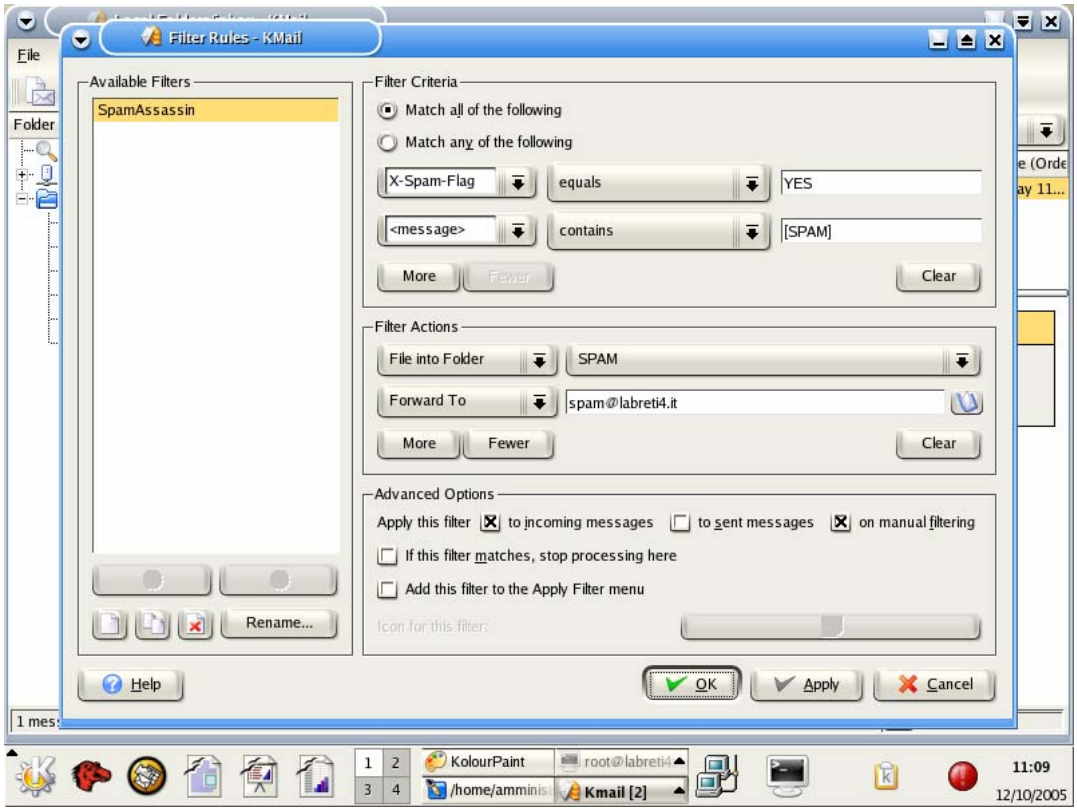
In questa si può vedere la configurazione degli account e nelle due successive quella dell'invio e della ricezione:







In quest'ultima figura invece si può notare la configurazione del filtro:



14.4 PROVE DI FUNZIONAMENTO

Vediamo ora una serie di prove che ho testato per verificare il funzionamento del mail server e di SpamAssassin. In questi esempi ho provato a mandare e-mail con l'utility `mail` oppure con il servizio `telnet` sulla porta 25 e ne ho verificato l'avvenuta ricezione nella mailbox relativa all'utente all'interno della cartella `/var/spool/mail/`.

1) Invio di un'e-mail da root ad un utente locale.

```
[root@labreti4 ~]# mail amministratore
Subject: prova ricezione attraverso pop

Funziona!!!!
.
Cc:
[root@labreti4 ~]#
```

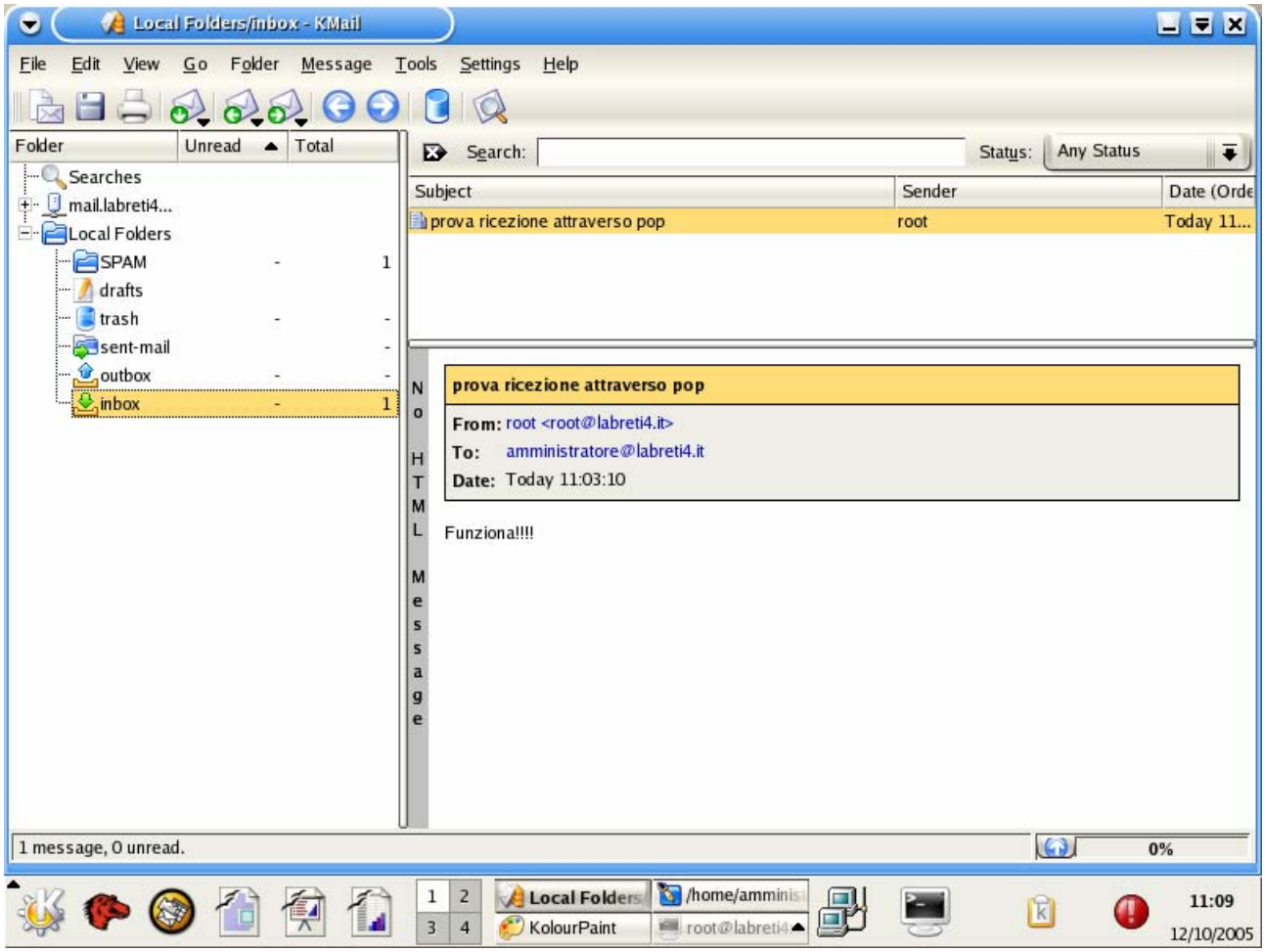
Verifico che l'e-mail non è considerata spam in quanto non è presente `X-Spam-Flag`, `X-Spam-Level` è vuoto e `X-Spam-Status` ha un punteggio di -100.6. Con `X-Spam-Checker-Version` si può vedere la versione di SpamAssassin e su quale dominio è attivo il servizio.

```
[root@labreti4 ~]# mail -f /var/spool/mail/amministratore
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/amministratore": 1 message 1 new
>N  1 root@labreti4.it      Wed Oct 12 10:46  19/702   "prova ricezione attra"
& 1
Message 1:
From root@labreti4.it  Wed Oct 12 10:46:06 2005
X-Original-To: amministratore
Delivered-To: amministratore@labreti4.it
To: amministratore@labreti4.it
Subject: prova ricezione attraverso pop
Date: Wed, 12 Oct 2005 10:46:06 +0200 (CEST)
From: root@labreti4.it (root)
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level:
X-Spam-Status: No, score=-100.6 required=5.0 tests=ALL_TRUSTED,AWL,
        NO_DNS_FOR_FROM,USER_IN_WHITELIST autolearn=disabled version=3.0.4
```

```
Funziona!!!!

& q
"/var/spool/mail/amministratore" complete
```

Vediamo in questo esempio la ricezione attraverso KMail che avviene con il protocollo POP.



2) Invio di un'email con mittente pluto@paperino.it il quale è inserito nella configurazione di `local.cf` come indirizzo in `blacklist_from`.

```
[root@labreti4 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
helo labreti4.it
250 mail.labreti4.it
mail from: <pluto@paperino.it>
```

```
250 Ok
rcpt to: <amministratore@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: prova da blacklist

Prova con indirizzo presente in blacklist
.
250 Ok: queued as 0862BC8515
quit
221 Bye
Connection closed by foreign host.
```

Verifico quindi che il messaggio è targato come spam. Sono presenti infatti le intestazioni `X-Spam-Flag: YES`, `X-Spam-Level` che contiene numerose "stelline" e quindi indicata un alto livello di spam raggiunto, `X-Spam-Status` con un punteggio addirittura di 98.9. Inoltre l'oggetto è stato riscritto aggiungendo la stringa `[SPAM]` all'inizio. Viene quindi riportata tutta l'analisi effettuata con i risultati e infine il messaggio.

```
[root@labreti4 ~]# mail -f /var/spool/mail/amministratore
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/amministratore": 1 message 1 new
>N  1 pluto@paperino.it     Wed Oct 12 10:28  68/2501  "[SPAM] prova da black"
& 1
Message 1:
From pluto@paperino.it  Wed Oct 12 10:28:48 2005
From: pluto@paperino.it
To: undisclosed-recipients:;
Subject: [SPAM] prova da blacklist
Date: Wed, 12 Oct 2005 10:27:41 +0200 (CEST)
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level: **********************************************
X-Spam-Status: Yes, score=98.9 required=5.0 tests=ALL_TRUSTED,NO_REAL_NAME,
        UNDISC_RECIPS,USER_IN_BLACKLIST autolearn=disabled version=3.0.4
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------=_434CC94A.F468D33E"

This is a multi-part message in MIME format.
```

```
------------=_434CC94A.F468D33E
Content-Type: text/plain
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "labreti4.it", has
identified this incoming email as possible spam.  The original message
has been attached to this so you can view it (if it isn't spam) or label
similar future email.  If you have any questions, see
the administrator of that system for details.

Content preview:  Prova con indirizzo presente in blacklist [...]

Content analysis details:   (98.9 points, 5.0 required)

 pts rule name              description
---- ---------------------- --------------------------------------------------
 0.2 NO_REAL_NAME           From: does not include a real name
 1.5 UNDISC_RECIPS          Valid-looking To "undisclosed-recipients"
 100 USER_IN_BLACKLIST      From: address is in the user's black-list
-2.8 ALL_TRUSTED            Did not pass through any untrusted hosts

safe    1



------------=_434CC94A.F468D33E
Content-Type: message/rfc822; x-spam-type=original
Content-Description: original message before SpamAssassin
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Return-Path: <pluto@paperino.it>
X-Original-To: amministratore@labreti4.it
Delivered-To: amministratore@labreti4.it
Received: from labreti4.it (labreti4.it [192.168.0.3])
        by mail.labreti4.it (Postfix) with SMTP id 0862BC8515
        for <amministratore@labreti4.it>; Wed, 12 Oct 2005 10:27:41 +0200 (CEST)
Subject: prova da blacklist
Message-Id: <20051012082741.0862BC8515@mail.labreti4.it>
Date: Wed, 12 Oct 2005 10:27:41 +0200 (CEST)
```

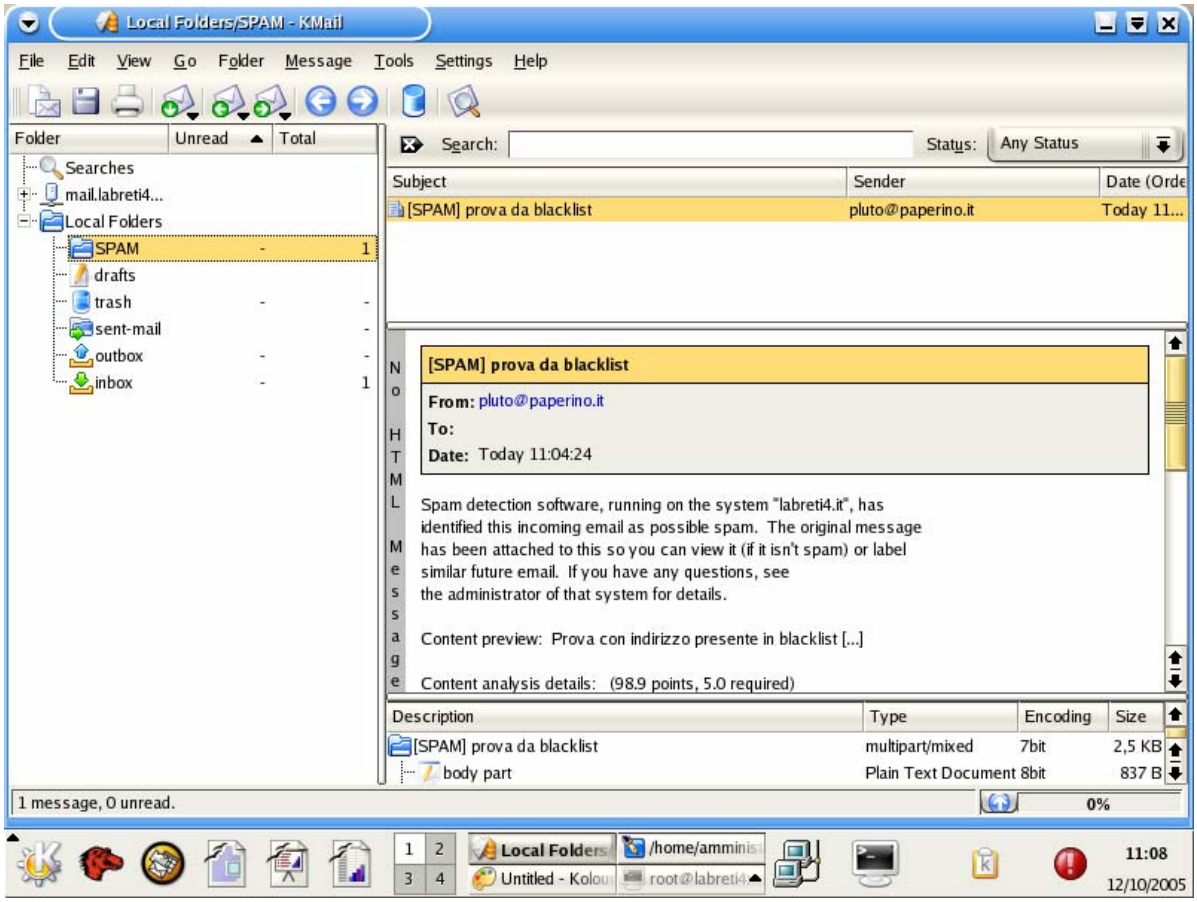
```
From: pluto@paperino.it
To: undisclosed-recipients:;

Prova con indirizzo presente in blacklist


------------=_434CC94A.F468D33E--

& q
"/var/spool/mail/amministratore" complete
```

Vediamo anche in questo esempio la ricezione con KMail e notiamo che il messaggio è stato spostato nella cartella SPAM.



3) Invio di e-mail da una macchina appartenente alla rete interna (il client con indirizzo IP192.168.1.2 e nome host `labreti5`) a un utente locale.

Dal client `labreti5` faccio partire il servizio telnet contattando il dominio `labreti4.it`.

```
[root@labreti5 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
helo labreti4.it
250 mail.labreti4.it
mail from: <root@labreti5>
250 Ok
rcpt to: <amministratore@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: prova da labreti5

prova da labreti5
.
250 Ok: queued as 0862BC8515
quit
221 Bye
Connection closed by foreign host.
```

Verifico che l'e-mail è arrivata all'utente locale.

```
[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 1 message 1 new
>N  1 root@labreti5.labret  Wed Oct 12 12:33  25/885   "prova da labreti5"
& 1
Message 1:
From root@labreti5.labreti4.it  Wed Oct 12 12:33:00 2005
X-Original-To: postino@labreti4.it
Delivered-To: postino@labreti4.it
From: root <root@labreti5.labreti4.it>
To: postino@labreti4.it
Subject: prova da labreti5
```

```
Date: Wed, 12 Oct 2005 11:31:00 +0200
User-Agent: KMail/1.7.1
MIME-Version: 1.0
Content-Type: text/plain;
  charset="us-ascii"
Content-Transfer-Encoding: 7bit
Content-Disposition: inline
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level:
X-Spam-Status: No, score=-1.7 required=5.0 tests=ALL_TRUSTED,NO_DNS_FOR_FROM
        autolearn=disabled version=3.0.4

prova da labreti5

& q
"/var/spool/mail/postino" complete
```

4) Invio di due e-mail. La prima con mittente un indirizzo (`aaa@labreti4.it`) appartenente al dominio `labreti4.it` che è stato inserito in `local.cf` come `whitelist_from` (`whitelist_from *@labreti4.it`). In questa e-mail nel corpo del messaggio è stata inserita la parola VIAGRA che nel file `local.cf` ha un punteggio di 7. Nella seconda invece lo stesso messaggio è mandato con mittente `aaa@bbb.it`.

```
[root@labreti4 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
mail from: <aaa@labreti4.it>
250 Ok
rcpt to: <postino@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: ciao

VIAGRA
.
250 Ok: queued as CB5B4C8D29
quit
```

```
221 Bye
Connection closed by foreign host.

[root@labreti4 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
mail from: <aaa@bbb.it>
250 Ok
rcpt to: <postino@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: ciao

VIAGRA
.
250 Ok: queued as 32DECC8D29
quit
221 Bye
Connection closed by foreign host.
```

Come si può notare il primo messaggio non viene targato come spam e raggiunge un punteggio di -91.5 e il secondo invece viene considerato spam raggiungendo il punteggio di 7.4. Il motivo è che nel primo caso il mittente dell'e-mail è un indirizzo presente in whitelist e ciò contribuisce notevolmente ad abbassare il punteggio.

```
[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 2 messages 2 new 2 unread
 U  1 aaa@labreti4.it       Wed Oct 12 12:04  21/791   "ciao"
>N  2 aaa@bbb.it            Wed Oct 12 12:06  71/2410  "[SPAM] ciao"
& 1
Message 1:
From aaa@labreti4.it  Wed Oct 12 12:04:20 2005
X-Original-To: postino@labreti4.it
Delivered-To: postino@labreti4.it
Subject: ciao
Date: Wed, 12 Oct 2005 12:03:42 +0200 (CEST)
From: aaa@labreti4.it
```

```
To: undisclosed-recipients:;
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level:
X-Spam-Status: No, score=-91.5 required=5.0 tests=ALL_TRUSTED,DRUGS_ERECTILE,
        DRUG_ED_CAPS,NO_DNS_FOR_FROM,NO_REAL_NAME,UNDISC_RECIPS,
        USER_IN_WHITELIST,VIAGRA autolearn=disabled version=3.0.4

VIAGRA

& 2
Message 2:
From aaa@bbb.it  Wed Oct 12 12:06:23 2005
From: aaa@bbb.it
To: undisclosed-recipients:;
Subject: [SPAM] ciao
Date: Wed, 12 Oct 2005 12:05:38 +0200 (CEST)
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level: *******
X-Spam-Status: Yes, score=7.4 required=5.0 tests=ALL_TRUSTED,DRUGS_ERECTILE,
        DRUG_ED_CAPS,NO_REAL_NAME,UNDISC_RECIPS,VIAGRA autolearn=disabled
        version=3.0.4
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------=_434CE022.41E045BF"

This is a multi-part message in MIME format.

------------=_434CE022.41E045BF
Content-Type: text/plain
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "labreti4.it", has
identified this incoming email as possible spam.  The original message
has been attached to this so you can view it (if it isn't spam) or label
similar future email.  If you have any questions, see
the administrator of that system for details.

Content preview:  VIAGRA [...]

Content analysis details:   (7.4 points, 5.0 required)
```

```
 pts rule name              description
---- ---------------------- --------------------------------------------------
 0.2 NO_REAL_NAME           From: does not include a real name
 1.5 UNDISC_RECIPS          Valid-looking To "undisclosed-recipients"
-2.8 ALL_TRUSTED            Did not pass through any untrusted hosts
 1.5 DRUG_ED_CAPS           BODY: Mentions an E.D. drug
 7.0 VIAGRA                 BODY: VIAGRA
 0.0 DRUGS_ERECTILE         Refers to an erectile drug

safe    1



------------=_434CE022.41E045BF
Content-Type: message/rfc822; x-spam-type=original
Content-Description: original message before SpamAssassin
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Return-Path: <aaa@bbb.it>
X-Original-To: postino@labreti4.it
Delivered-To: postino@labreti4.it
Received: from labreti4.it (labreti4.it [192.168.0.3])
        by mail.labreti4.it (Postfix) with SMTP id 32DECC8D29
        for <postino@labreti4.it>; Wed, 12 Oct 2005 12:05:38 +0200 (CEST)
Subject: ciao
Message-Id: <20051012100538.32DECC8D29@mail.labreti4.it>
Date: Wed, 12 Oct 2005 12:05:38 +0200 (CEST)
From: aaa@bbb.it
To: undisclosed-recipients:;

VIAGRA



------------=_434CE022.41E045BF--

& q
"/var/spool/mail/postino" complete
```

5) Invio di un'e-mail contenente la stringa TONER che in `local.cf` ha un punteggio di 2.5

```
[root@labreti4 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
mail from: <aaa@bbb.it>
250 Ok
rcpt to: <postino@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: ciao

TONER
.
250 Ok: queued as D01D8C8D29
quit
221 Bye
Connection closed by foreign host.
```

Notiamo che il punteggio assegnato non è sufficiente per timbrare il messaggio come spam. Il punteggio raggiunto è infatti di 1.9 in quanto intervengono anche altri tests.

```
[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 1 message 1 new
>N  1 aaa@bbb.it            Wed Oct 12 15:01  19/695   "ciao"
& 1
Message 1:
From aaa@bbb.it  Wed Oct 12 15:01:01 2005
X-Original-To: postino@labreti4.it
Delivered-To: postino@labreti4.it
Subject: ciao
Date: Wed, 12 Oct 2005 15:00:28 +0200 (CEST)
From: aaa@bbb.it
To: undisclosed-recipients:;
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level: *
X-Spam-Status: No, score=1.9 required=5.0 tests=ALL_TRUSTED,AWL,NO_REAL_NAME,
```

```
        UNDISC_RECIPS autolearn=disabled version=3.0.4

TONER

& q
"/var/spool/mail/postino" complete
```

6) Invio di due e-mail. Nella prima nell'oggetto la stringa [OT] ; nel file `local.cf` è presente una regola che indica che in questo caso viene assegnato un punteggio di 3.5. Nella seconda riporto lo stesso messaggio preoccupandomi però prima di inviare il messaggio di alzare il punteggio a 7.

```
[root@labreti4 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
mail from: <aaa@bbb.it>
250 Ok
rcpt to: <postino@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: [OT] ciao

ciao
.
250 Ok: queued as 2DF69C8D29
quit
221 Bye
Connection closed by foreign host.
```

Notiamo che nella prima e-mail il punteggio raggiunto è di 4.9 quindi non sufficiente a targare l'e-mail come spam.

```
[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 1 message 1 new
>N  1 aaa@bbb.it            Wed Oct 12 13:05  19/715   "[OT] ciao"
& 1
Message 1:
```

```
From aaa@bbb.it  Wed Oct 12 13:05:50 2005
X-Original-To: postino@labreti4.it
Delivered-To: postino@labreti4.it
Subject: [OT] ciao
Date: Wed, 12 Oct 2005 13:04:29 +0200 (CEST)
From: aaa@bbb.it
To: undisclosed-recipients:;
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level: ****
X-Spam-Status: No, score=4.9 required=5.0 tests=ALL_TRUSTED,ANTI_TROLL,AWL,
        NO_REAL_NAME,UNDISC_RECIPS autolearn=disabled version=3.0.4

ciao

& q
"/var/spool/mail/postino" complete
```

Proviamo a mandare la seconda e-mail dopo aver modificato il punteggio.

```
root@labreti4 ~]# telnet labreti4.it 25
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
220 mail.labreti4.it ESMTP Postfix
mail from: <aaa@bbb.it>
250 Ok
rcpt to: <postino@labreti4.it>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: [OT] riprova

ciao
.
250 Ok: queued as D2D17C8D29
quit
221 Bye
Connection closed by foreign host.
```

Notiamo che questa volta il messaggio è stato targato come spam raggiungendo un punteggio di 5.3.

```
[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 2 messages 1 new
    1 aaa@bbb.it            Wed Oct 12 13:05  20/726   "[OT] ciao"
>N  2 aaa@bbb.it            Wed Oct 12 13:13  69/2369  "[SPAM] [OT] riprova"
& 2
Message 2:
From aaa@bbb.it  Wed Oct 12 13:13:42 2005
From: aaa@bbb.it
To: undisclosed-recipients:;
Subject: [SPAM] [OT] riprova
Date: Wed, 12 Oct 2005 13:13:07 +0200 (CEST)
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level: *****
X-Spam-Status: Yes, score=5.3 required=5.0 tests=ALL_TRUSTED,ANTI_TROLL,AWL,
        NO_REAL_NAME,UNDISC_RECIPS autolearn=disabled version=3.0.4
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------=_434CEFE9.8A24FE6C"

This is a multi-part message in MIME format.

------------=_434CEFE9.8A24FE6C
Content-Type: text/plain
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "labreti4.it", has
identified this incoming email as possible spam.  The original message
has been attached to this so you can view it (if it isn't spam) or label
similar future email.  If you have any questions, see
the administrator of that system for details.

Content preview:  ciao [...]

&
At EOF
& q
"/var/spool/mail/postino" complete
```

```
[root@labreti4 ~]# mail -f /var/spool/mail/postino
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/postino": 7 messages
>   1 root@labreti5.labret  Wed Oct 12 12:33  26/896   "prova da labreti5"
    2 root@labreti5.labret  Wed Oct 12 12:36  26/912   "test da labreti5"
    3 aaa@bbb.it            Wed Oct 12 13:05  20/726   "[OT] ciao"
    4 aaa@bbb.it            Wed Oct 12 13:08  20/726   "[OT] ciao"
    5 aaa@bbb.it            Wed Oct 12 13:10  20/726   "[OT] ciao"
    6 aaa#bbb.it@labreti4.  Wed Oct 12 13:12  21/792   "[OT] ciao"
    7 aaa@bbb.it            Wed Oct 12 13:13  70/2380  "[SPAM] [OT] riprova"
& 7
Message 7:
From aaa@bbb.it  Wed Oct 12 13:13:42 2005
From: aaa@bbb.it
To: undisclosed-recipients:;
Subject: [SPAM] [OT] riprova
Date: Wed, 12 Oct 2005 13:13:07 +0200 (CEST)
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level: *****
X-Spam-Status: Yes, score=5.3 required=5.0 tests=ALL_TRUSTED,ANTI_TROLL,AWL,
        NO_REAL_NAME,UNDISC_RECIPS autolearn=disabled version=3.0.4
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----------=_434CEFE9.8A24FE6C"

This is a multi-part message in MIME format.

------------=_434CEFE9.8A24FE6C
Content-Type: text/plain
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "labreti4.it", has
identified this incoming email as possible spam.  The original message
has been attached to this so you can view it (if it isn't spam) or label
similar future email.  If you have any questions, see
the administrator of that system for details.

Content preview:  ciao [...]

Content analysis details:   (5.3 points, 5.0 required)
```

```
 pts rule name              description
---- ---------------------- --------------------------------------------------
 0.2 NO_REAL_NAME           From: does not include a real name
 1.5 UNDISC_RECIPS          Valid-looking To "undisclosed-recipients"
 7.0 ANTI_TROLL             Subject: found OT tag
-2.8 ALL_TRUSTED            Did not pass through any untrusted hosts
-0.5 AWL                    AWL: From: address is in the auto white-list

safe    1



------------=_434CEFE9.8A24FE6C
Content-Type: message/rfc822; x-spam-type=original
Content-Description: original message before SpamAssassin
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Return-Path: <aaa@bbb.it>
X-Original-To: postino@labreti4.it
Delivered-To: postino@labreti4.it
Received: from labreti4.it (labreti4.it [192.168.0.3])
        by mail.labreti4.it (Postfix) with SMTP id D2D17C8D29
        for <postino@labreti4.it>; Wed, 12 Oct 2005 13:13:07 +0200 (CEST)
Subject: [OT] riprova
Message-Id: <20051012111307.D2D17C8D29@mail.labreti4.it>
Date: Wed, 12 Oct 2005 13:13:07 +0200 (CEST)
From: aaa@bbb.it
To: undisclosed-recipients:;

ciao



------------=_434CEFE9.8A24FE6C--

& q
"/var/spool/mail/postino" complete
```

7) Ricezione di e-mail attraverso l'apertura di una sessione POP con telnet sulla porta 110.

```
[root@labreti4 ~]# telnet labreti4.it 110
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
+OK dovecot ready.
USER postino
+OK
PASS postino
+OK Logged in.
LIST
+OK 9 messages:
1 1123
2 766
3 769
4 810
5 824
6 2690
7 778
8 764
9 769
.
RETR  9
+OK 769 octets
Return-Path: <root@labreti4.it>
X-Original-To: postino
Delivered-To: postino@labreti4.it
Received: by mail.labreti4.it (Postfix, from userid 0)
        id B0C03C8D31; Mon, 17 Oct 2005 09:41:38 +0200 (CEST)
To: postino@labreti4.it
Subject: ciao
Message-Id: <20051017074138.B0C03C8D31@mail.labreti4.it>
Date: Mon, 17 Oct 2005 09:41:38 +0200 (CEST)
From: root@labreti4.it (root)
X-Spam-Checker-Version: SpamAssassin 3.0.4 (2005-06-05) on labreti4.it
X-Spam-Level:
X-Spam-Status: No, score=-99.8 required=5.0 tests=ALL_TRUSTED,AWL,
        NO_DNS_FOR_FROM,USER_IN_WHITELIST autolearn=disabled version=3.0.4
Status: RO
X-UID: 17
Content-Length: 7
X-Keywords:
```

```
ciao
.
quit
+OK Logging out.
Connection closed by foreign host.
```

8) Ricezione di e-mail attraverso l'apertura di una sessione IMAP con telnet sulla porta 143.

```
[root@labreti4 ~]# telnet labreti4.it 143
Trying 192.168.0.3...
Connected to labreti4.it (192.168.0.3).
Escape character is '^]'.
* OK dovecot ready.
a100 LOGIN postino postino
a100 OK Logged in.
a101 SELECT inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags
permitted.
* 9 EXISTS
* 0 RECENT
* OK [UNSEEN 1] First unseen.
* OK [UIDVALIDITY 1129119996] UIDs valid
* OK [UIDNEXT 18] Predicted next UID
a101 OK [READ-WRITE] Select completed.
a102 FETCH 1:2 (flags body[header.fields (subject)])
* 1 FETCH (FLAGS (\Seen) BODY[HEADER.FIELDS (SUBJECT)] {17}
Subject: ciao
)
* 2 FETCH (FLAGS (\Seen) BODY[HEADER.FIELDS (SUBJECT)] {17}
Subject: test
)
a102 OK Fetch completed.
a103 fetch 1 (body[text])
* 1 FETCH (BODY[TEXT] {17}
ciao come stai?
a103 OK Fetch completed.
a104 LOGOUT
* BYE Logging out
a104 OK Logout completed.
Connection closed by foreign host.
```

15. CONCLUSIONI

In questa tesi abbiamo visto e approfondito la tematica dello spamming, fenomeno presente da anni nella rete e a cui è difficile sfuggire a causa dell'ottima organizzazione di chi lo pratica. Abbiamo analizzato anche diverse tecniche di difesa, ma fino a quando non verranno presi provvedimenti da un consistente numero di provider è difficile che il problema diminuisca. Indubbiamente abbiamo notato che qualcosa si sta iniziando a muovere con la recentissima tecnica del DKIM (Domain Keys Identified Mail) ma la strada è ancora lunga.

Ancora adesso l'onere di questa pratica intrusiva è scaricato sugli utenti che devono attrezzarsi al meglio. Per questo motivo in questa tesi ho voluto riportare una vasta panoramica sugli strumenti adatti allo scopo che offrono una sorta di tutela ma non eliminano il problema.

Si potrebbe pensare che tutto questo possa essere evitato se Internet fosse più ordinata e regolamentata, se ci fossero sistemi di autenticazione sicuri, se ci fossero più controlli sui dati trasmessi sulle linee. Tutto ciò però ha il suo rovescio della medaglia: queste misure segnerebbero la fine della libertà di cui è possibile godere su Internet, la quale comporta comunque responsabilità e doveri per una convivenza civile.

La soluzione migliore sarebbe eliminare il problema alla fonte, purtroppo però questo risulta essere molto difficile, quasi impossibile per l'abilità con cui gli spammer si nascondono e con cui creano nuove tecniche sempre più efficaci per ovviare agli ostacoli che gli si presentano. Per questo motivo si può solo arrivare ad un compromesso che consiste nel limitare lo spam, in attesa che lo sforzo collaborativo di tutti dia i suoi buoni frutti.

16. APPENDICE A – Provvedimento del Garante della privacy

Provvedimento generale

GARANTE PER LA PROTEZIONE DEI DATI PERSONALI

Nella riunione odierna, in presenza del prof. Stefano Rodotà, presidente, del prof. Giuseppe Santaniello, vice-presidente, del prof. Gaetano Rasi e del dr. Mauro Paissan, componenti e del dott. Giovanni Buttarelli, segretario generale;

VISTI i reclami e le segnalazioni pervenuti all'Autorità circa l'indebito utilizzo della posta elettronica per finalità promozionali e pubblicitarie;

VISTE le decisioni già adottate dal Garante in materia e ritenuto necessario adottare un provvedimento di carattere generale sull'applicazione della disciplina in materia;

VISTI la legge 31 dicembre 1996, n. 675, il d.lg. 13 maggio 1998, n. 171 e le altre disposizioni applicabili;

VISTI gli atti d'ufficio;

VISTE le osservazioni formulate dal segretario generale ai sensi dell'art. 15 del regolamento del Garante n. 1/2000;

RELATORE il dott. Mauro Paissan;

PREMESSO:

1. I DISAGI DI NUMEROSI UTENTI

Continuano a pervenire a questa Autorità diverse centinaia di reclami e segnalazioni da parte di utenti di reti telematiche e di associazioni per la tutela dei diritti di utenti e consumatori, che contestano la ricezione di messaggi di posta elettronica per scopi promozionali, pubblicitari, di informazione commerciale o di vendita diretta, inviati senza che gli interessati abbiano manifestato in precedenza il proprio consenso informato.

Numerosi interessati espongono anche ulteriori disagi derivanti dalla costante ripetizione di analoghi messaggi da parte di uno stesso mittente titolare del trattamento, dai vani tentativi esperiti per ottenere sia la cancellazione del proprio indirizzo di posta elettronica presso i mittenti, sia l'interruzione di altri messaggi. Altre segnalazioni riguardano gli inconvenienti che derivano dalla ricezione di e-mail anonime o prive dell'indicazione di un indirizzo, oppure delle coordinate veritiere di un reale mittente.

Nella prevalenza dei casi, agli interessati non è stato previamente richiesto, come dovuto, uno specifico consenso preceduto da un'idonea informativa che illustri adeguatamente le modalità e le caratteristiche dei messaggi.

In altri casi i messaggi sono inviati da imprese -anche in questo caso senza consenso- per promuovere, presso clienti, prodotti o servizi analoghi a quelli forniti in un rapporto contrattuale, oppure per offrire altri tipi di prodotti o servizi distribuiti anche da terzi.

Il Garante ha fornito assistenza a numerosi cittadini, indicando le opportune modalità di tutela; ha poi attivamente cooperato in sede comunitaria per l'adozione di decisioni comuni alle autorità di garanzia dei Paesi dell'Unione europea, pubblicate nel sito Internet di quest'ultima e in quello del Garante (www.garanteprivacy.it).

L'Autorità ha anche accolto numerosi ricorsi (art. 29 legge n. 675/1996), a seguito dei quali sono stati impartiti specifici divieti di trattamento dei dati. Sono stati altresì avviati i procedimenti per applicare le pertinenti sanzioni amministrative e sono stati trasmessi gli atti all'autorità giudiziaria penale nei casi in cui erano configurabili reati.

Con la collaborazione di forze di polizia, incaricate da questa Autorità di svolgere i necessari controlli e di dare esecuzione ai provvedimenti, sono stati eseguiti in loco, presso fornitori di servizi ed altri titolari di trattamento, vari provvedimenti di sospensione temporanea di ogni operazione illecita del trattamento dei dati personali da parte di società risultate responsabili di attività svolte in modo sistematico. Infine, sono stati eseguiti accertamenti presso altri fornitori di servizi di accesso ad Internet o ulteriori soggetti, per verificare la rispondenza dei trattamenti di dati alla normativa vigente.

A conclusione di queste attività, il Garante ravvisa la necessità di adottare un provvedimento di carattere generale per indicare le misure che gli operatori del settore devono adottare al fine di conformarsi alla disciplina generale sull'uso dei dati personali, specie nel settore delle

comunicazioni (in particolare, alla legge 31 dicembre 1996, n. 675, al decreto legislativo 13 maggio 1998, n. 171 e al decreto legislativo 22 maggio 1999, n. 185). L'Autorità ritiene inoltre necessario inibire il trattamento illecito di dati risultante da altre segnalazioni il cui esame è stato riunito in un unico procedimento, in particolare di quelle relative a titolari di trattamento identificabili.

2. INVIO LECITO DI POSTA ELETTRONICA PUBBLICITARIA

Gli indirizzi di posta elettronica recano dati di carattere personale da trattare nel rispetto della normativa in materia (art. 1, comma 1 lett. c), legge n. 675).

La loro utilizzazione per scopi promozionali e pubblicitari è possibile solo se il soggetto cui riferiscono i dati ha manifestato in precedenza un consenso libero, specifico e informato.

Il consenso è necessario anche quando gli indirizzi sono formati ed utilizzati automaticamente con un software senza l'intervento di un operatore, o in mancanza di una previa verifica della loro attuale attivazione o dell'identità del destinatario del messaggio, e anche quando gli indirizzi non sono registrati dopo l'invio dei messaggi.

Questo assetto, basato su una scelta dell'interessato c.d. di opt-in, è stato ribadito nel 1998 (con il d.lg. n. 171) prima ancora che una recente direttiva comunitaria lo estendesse a tutti i Paesi dell'Unione europea (n. 2002/58/CE in fase di recepimento in Italia, pubblicata sulla G.U.C.E. n. L 201 del 31 luglio 2002).

Questa Autorità si è pronunciata più volte in materia ribadendo che la circostanza che gli indirizzi di posta elettronica possano essere reperiti con una certa facilità in Internet non comporta il diritto di utilizzarli liberamente per inviare messaggi pubblicitari (cfr., tra l'altro, la decisione dell'11 gennaio 2001 - in Bollettino del Garante n. 16).

In particolare, i dati dei singoli utenti che prendono parte a gruppi di discussione in Internet sono resi conoscibili in rete per le sole finalità di partecipazione ad una determinata discussione e non possono essere utilizzati per fini diversi qualora manchi un consenso specifico (art. 9, comma 1, lettere a) e b), legge n. 675).

Ad analoga conclusione deve pervenirsi per gli indirizzi di posta elettronica compresi nella lista "anagrafica" degli abbonati ad un Internet provider (qualora manchi, anche in questo caso, un consenso libero e specifico), oppure pubblicati su siti web di soggetti pubblici per fini istituzionali.

Tali considerazioni valgono anche con riferimento ai messaggi pubblicitari inviati a gestori di siti web -anche di soggetti privati- utilizzando gli indirizzi pubblicati sugli stessi siti, o che sono reperibili consultando gli elenchi dei soggetti che hanno registrato i nomi a dominio. In quest'ultimo caso, infatti, la conoscibilità in rete degli indirizzi è volta a identificare il soggetto che è o appare responsabile, sul piano tecnico o amministrativo, di un nome a dominio o di altre funzioni rispetto a servizi Internet (per la tutela di vari diritti sul piano civile e penale, anche ai sensi della legge n. 675) e non anche a rendere l'interessato disponibile all'invio di messaggi pubblicitari).

In tutti questi casi, l'utilizzo spesso massiccio della posta elettronica comporta una lesione ingiustificata dei diritti dei destinatari, costretti ad impiegare diverso tempo per mantenere un collegamento e per ricevere, come pure per esaminare e selezionare, tra i diversi messaggi ricevuti, quelli attesi o ricevibili, nonché a sostenere i correlativi costi per il collegamento telefonico (incrementati anche da messaggi di dimensioni rilevanti che rallentano tali operazioni), oppure ad adottare "filtri", a verificare più attentamente la presenza di virus, o a cancellare rapidamente materiali inadatti a minori specie in ambito domestico.

Il fenomeno interessa anche piccole e grandi imprese destinatarie di un elevato numero di messaggi, le quali devono farsi carico di misure interne e di costi anche organizzativi per contrastarlo.

Questo ingiustificato riversamento sugli utenti dei costi pubblicitari si verifica anche relativamente a messaggi inviati da singole persone fisiche che, in vari casi esaminati, non si limitano ad una comunicazione episodica, ma intraprendono una comunicazione sistematica per fini personali o, addirittura, una diffusione di dati cui è applicabile la disciplina in materia di protezione dei dati personali (art. 3 legge n. 675).

3. IL QUADRO GIURIDICO SU INFORMATIVA E CONSENSO

La legge individua il contenuto dell'informativa agli interessati, nonché i casi in cui è necessario il consenso espresso dell'interessato o è possibile prescinderne (artt. 10, 11, 12 e 20 legge n. 675).

Al riguardo va nuovamente rilevato che non può farsi a meno del consenso ritenendo che i dati personali relativi all'indirizzo di posta elettronica –e all'indirizzo in particolare- siano "pubblici" in quanto conoscibili da chiunque.

Le disposizioni normative che si riferiscono a questo aspetto (artt. 12, comma 1, lett. c) e 20, comma 1, lett. b) legge cit.) sono infatti applicabili solo quando vi è un pubblico registro, elenco,

atto o documento conoscibile da chiunque perché vi è una specifica disciplina che ne impone la conoscibilità indifferenziata da parte del pubblico, e non anche quando i dati personali sono conoscibili da chiunque per mere circostanze di fatto (si pensi, oltre ai casi già richiamati di raccolta su siti web o di messaggi trasmessi su newsgroup o su mailing list, agli indirizzi di posta elettronica raccolti in rete tramite appositi software o mediante comuni motori di ricerca).

Il principio del consenso è quindi già operante nel nostro ordinamento prima ancora di essere affermato senza eccezioni su scala europea, dalla menzionata direttiva n. 2002/58 in fase di recepimento, a tutta la posta elettronica comunque inviata per fini di commercializzazione diretta (si vedano in particolare l'art. 13 e il considerando n. 40).

Il quadro evidenziato trova conferma nella disciplina sulla protezione dei consumatori nei contratti a distanza che, in riferimento al rapporto sottostante ai fini del quale si procede al trattamento di dati personali, vieta ai fornitori l'impiego della posta elettronica in mancanza del consenso preventivo del consumatore, in relazione a determinati scopi tra i quali rientrano anche quelli pubblicitari (art. 10, comma 1, d.lg. 22 maggio 1999, n. 185).

Per gli aspetti relativi alla protezione dei dati personali non devono essere peraltro considerate le disposizioni del recente decreto legislativo 9 aprile 2003, n. 70, sul commercio elettronico, dichiarate in proposito espressamente inapplicabili (art. 1, comma 2, lett. b) d.lg. n. 70 cit.).

Il consenso, da documentare per iscritto, deve essere manifestato liberamente, in modo esplicito e in forma differenziata rispetto alle diverse finalità e alle categorie di servizi e prodotti offerti, prima dell'inoltro dei messaggi (art. 11 legge n. 675).

Tale disciplina non può essere elusa inviando una prima e-mail che, nel chiedere un consenso abbia comunque un contenuto promozionale oppure pubblicitario, oppure riconoscendo solo un diritto di tipo c.d. "opt-out" al fine di non ricevere più messaggi dello stesso tenore.
Al contrario, è opportuna e va incoraggiata la prassi di alcuni fornitori i quali, dopo aver ottenuto realmente un valido consenso dei destinatari, danno semplice conferma della sua manifestazione, attraverso un messaggio volto unicamente ad annunciare il successivo inoltro di materiale pubblicitario. Tale prassi, se utilizzata correttamente, consente tra l'altro di verificare l'effettiva corrispondenza dell'indirizzo di posta elettronica ai soggetti che avevano espresso il consenso, nonché di accertare il permanere di tale volontà.

L'insieme dei diritti riconosciuti dalla legge agli utenti determina, in caso di loro violazione, un trattamento illecito dei dati che:

- è già vietato direttamente dalla legge, senza che sia necessario adottare uno specifico provvedimento interdittivo del Garante dell'autorità giudiziaria;- determina, a seconda dei casi, l'applicazione di sanzioni amministrative pecuniarie, in particolare per
- omessa informativa od omessa notificazione (artt. 10, 34 e 39 legge n. 675; art. 12 d.lg. n. 185/1999);
- comporta il rimborso delle spese e dei diritti relativi al procedimento attivato da un fondato ricorso al Garante, oppure da un'azione dinanzi al giudice civile, come pure il risarcimento dei danni, specie di tipo patrimoniale, che derivino dai fatti illeciti e siano comprovati dall'interessato in relazione ai disagi sopra illustrati;
- rende applicabile anche una sanzione penale qualora il trattamento illecito dei dati sia effettuato al fine di trarne per sé o per altri un profitto o per arrecare ad altri un danno, con la pena accessoria della pubblicazione della sentenza di condanna (artt. 35 e 38 legge n. 675).

4. MESSAGGI PUBBLICITARI A PROPRI CLIENTI

Per effetto del recepimento della direttiva 2002/58/CE sarà peraltro possibile integrare, nel prossimo futuro, la disciplina sopra illustrata, permettendo a talune società di far conoscere a propri clienti prodotti o servizi analoghi a quelli per i quali si è già stabilito un rapporto, con i medesimi clienti, di vendita di prodotti o servizi.

In tali casi, la società titolare del trattamento (dopo aver informato preventivamente e adeguatamente il cliente) potrà procedere all'invio del messaggio pubblicitario, offrendo però al cliente, in modo chiaro e distinto (sia al momento della raccolta dei suoi dati, sia in occasione di ciascun messaggio) il diritto di rifiutare sin dall'inizio tale uso dei dati o di obiettare, gratuitamente e in maniera agevole, anche successivamente (art. 13, par. 2, direttiva n. 2002/58/CE cit.)

5. MESSAGGI PER CONTO TERZI E ACQUISTO DI BANCHE DATI

In alcuni casi portati all'attenzione del Garante, l'invio di messaggi pubblicitari era stato effettuato, per conto di terzi committenti, da società specializzate che utilizzano indirizzi di posta elettronica contenuti in proprie banche dati.

Tali società, da considerarsi "titolari" o contitolari del trattamento dei dati a seconda del rapporto che si instaura con il committente e delle modalità di concreta utilizzazione dei dati, sono tenute a rispettare le disposizioni in tema di informativa e specifico consenso, anche per quanto riguarda l'eventuale comunicazione di dati personali ai committenti medesimi e le relative finalità.

Ciò comporta un quadro di obblighi e possibili responsabilità anche penali che gli operatori devono verificare con attenzione, anche quando la società specializzata incaricata sia stabilita fuori dell'Unione europea.

Dall'esame dei reclami e delle segnalazioni pervenuti al Garante è risultato, altresì, che alcuni dei soggetti che hanno utilizzato la posta elettronica per l'invio di messaggi pubblicitari avevano acquisito da terzi le banche dati contenenti gli indirizzi dei destinatari. In questi casi, chi acquisisce la banca dati deve accertare che ciascun interessato abbia validamente acconsentito alla comunicazione del proprio indirizzo di posta elettronica ed al suo successivo utilizzo ai fini di invio di materiale pubblicitario; al momento in cui registra i dati deve poi inviare in ogni caso, a tutti gli interessati, un messaggio di informativa che precisi gli elementi indicati nell'art. 10 della legge n. 675, comprensivi di un riferimento di luogo -e non solo di posta elettronica- presso cui l'interessato possa esercitare i diritti riconosciuti dalla legge.

6. DIRITTI DEGLI INTERESSATI

Indipendentemente dal rapporto esistente tra i mittenti ed i destinatari dei messaggi, chi detiene i dati deve assicurare in ogni caso agli interessati la possibilità di far valere in ogni momento i diritti riconosciuti dalla legge, i quali sono spesso esercitati per conoscere da quale fonte sono stati tratti i dati, o per far interrompere gratuitamente la loro ulteriore utilizzazione ai fini commerciali-pubblicitari, oppure per far cancellare i dati trattati in violazione di legge (art. 13, comma 1, lett. e), della legge).

Nel sito Internet del Garante è riportato un modello-tipo per esercitare tali diritti in maniera agevole, gratuitamente e senza particolari formalità, anche verbalmente o mediante posta elettronica, dimostrando la propria identità (art. 17, comma 1, d.P.R. n. 501 del 31 marzo 1998). Tale modello è utilizzabile in luogo di altri reperibili in reti telematiche che non sono pienamente validi in quanto si riferiscono anche ad aspetti non riconosciuti dall'art. 13 della legge n. 675 (ad esempio, chiedono il rilascio di attestazioni o la copia di autorizzazioni non previste).

I diritti vanno esercitati sulla base di tale modello direttamente presso l'indirizzo conoscibile del titolare o del responsabile del trattamento, riservando solo ad un'eventuale momento successivo l'instaurazione di una procedura contenziosa dinanzi al Garante o all'autorità giudiziaria.

Anche ai fini dell'esercizio di tali diritti, deve ritenersi che l'invio anonimo di messaggi pubblicitari senza l'indicazione di un mittente identificabile concreti già oggi un trattamento illecito di dati personali, a prescindere da quanto dispone il citato d.lg. n. 70/2003 sul commercio elettronico (come si è visto, fuori della materia della protezione dei dati personali) e da quanto, in riferimento ai dati personali, sarà previsto con il recepimento della direttiva n. 2002/58/CE (la quale non consente l'invio di messaggi pubblicitari quando l'identità del mittente viene camuffata o addirittura celata e quando non viene fornito un indirizzo valido che consenta al destinatario di richiedere la cessazione delle comunicazioni: art. 13, par. 4, dir. cit.).

I mittenti dei messaggi devono quindi indicare già oggi, in modo chiaro, la fonte di provenienza del messaggio, nonché il soggetto e l'indirizzo –non solo di posta elettronica- presso cui i destinatari possono esercitare i propri diritti (si veda, in proposito, l'art. 10, comma 1, lett. f) della legge n. 675). Appare altresì conforme al principio di correttezza indicare nell'oggetto del messaggio la sua tipologia pubblicitaria-commerciale (art. 9, comma 1, lett. a), legge n. 675).

7. ELENCHI DI POSSIBILI DESTINATARI

L'eventuale elenco predisposto da operatori, contenente i nominativi dei soggetti che non hanno manifestato il consenso o che lo hanno revocato (c.d. black list) non può essere utilizzato per porre a carico degli interessati, anche indirettamente, un onere di iscrizione nell'elenco medesimo.

Come si è illustrato, il consenso ha un connotato autorizzatorio "positivo" in base al quale l'eventuale silenzio dell'interessato comporta il diniego del consenso eventualmente richiesto e non rileva come assenso tacito all'invio dei messaggi.

Consta peraltro che alcuni operatori intendono adottare la diversa prassi di redigere anche tramite siti web appositi elenchi di persone che hanno manifestato il consenso, distinti in base alle diverse categorie di messaggi commerciali-pubblicitari che gli interessati hanno acconsentito a ricevere. Tale prassi, se correttamente seguita, può rappresentare una misura utile, sul piano organizzativo, per garantire un più effettivo rispetto della volontà espressa dai singoli. A tale riguardo, costituirà una pratica utile quella di garantire agli interessati la possibilità di inserire direttamente il proprio nome nelle diverse liste o di cancellarlo dalle stesse, magari attraverso un'apposita pagina web, ferma restando l'esigenza di identificarli.

8. E-MAIL PROVENIENTI DALL'ESTERO

Ad alcuni messaggi, in quanto provenienti dall'estero, non è applicabile la legge italiana sulla protezione dei dati personali.

Ciò non comporta l'assoluta mancanza di rimedi o tutela, potendo l'utente chiedere una verifica da parte della competente autorità nazionale di protezione dei dati personali, ove istituita nel Paese eventualmente individuabile dal messaggio.

In altri casi, come quelli relativi alle leggi degli stati federali, l'invio di messaggi pubblicitari di posta elettronica può essere illecito in base alla legge di alcuni stati, per cui è parimenti possibile, per gli utenti, chiedere alle competenti autorità pubbliche degli stati di valutare la perseguibilità degli illeciti.

Va infine tenuto presente che alcune e-mail indesiderate possono essere lo strumento per commettere reati comuni (ad esempio di truffa) che devono considerarsi commessi nel territorio italiano quando, sebbene l'azione è avvenuta all'estero, l'evento-reato che ne deriva si è verificato in Italia.

Questa Autorità si riserva di valutare la posizione dei singoli fornitori di servizi i cui trattamenti sono stati oggetto di segnalazione, anche alla luce dell'ulteriore documentazione eventualmente pervenuta.

In questo quadro, con separati provvedimenti relativi all'esame dei singoli reclami e segnalazioni, si provvederà, oltre alle eventuali trasmissioni di atti all'autorità giudiziaria penale:

a) a contestare la violazione amministrativa relativa agli obblighi di informativa di cui all'art. 10 della legge 31 dicembre 1966, n. 675;

b) ad avviare il procedimento per l'applicazione delle ulteriori sanzioni amministrative previste dal d.lg. n. 185/1999;

TUTTO CIÒ PREMESSO IL GARANTE:

1. ai sensi dell'art. 31, comma 1, lett. l) della legge 31 dicembre 1996, n. 675, vieta l'ulteriore trattamento illecito di dati personali realizzato a scopi di invio di materiale pubblicitario o di vendita diretta, ovvero per il compimento di ricerche di mercato o di comunicazione

commerciale interattiva, effettuato in violazione delle disposizioni sopra richiamate da parte dei soggetti cui si riferiscono le segnalazioni e i reclami pervenuti;

2. ai sensi dell'art. 31, comma 1, lett. c) della legge 31 dicembre 1996, n. 675, segnala ai titolari del trattamento di cui agli atti del procedimento la necessità di conformare i trattamenti di dati personali ai principi richiamati nel presente provvedimento.

Roma, 29 maggio 2003

IL PRESIDENTE
Rodotà

IL RELATORE
Paissan

IL SEGRETARIO GENERALE
Buttarelli

17. APPENDICE B – Files di configurazione principali

Postfix

File di configurazione di Postfix `/etc/postfix/main.cf`

```
# Global Postfix configuration file. This file lists only a subset
# of all 300+ parameters. See the postconf(5) manual page for a
# complete list.
#
# The general format of each line is: parameter = value. Lines
# that begin with whitespace continue the previous line. A value can
# contain references to other $names or ${name}s.
#
# NOTE - CHANGE NO MORE THAN 2-3 PARAMETERS AT A TIME, AND TEST IF
# POSTFIX STILL WORKS AFTER EVERY CHANGE.

# SOFT BOUNCE
#
# The soft_bounce parameter provides a limited safety net for
# testing.  When soft_bounce is enabled, mail will remain queued that
# would otherwise bounce. This parameter disables locally-generated
# bounces, and prevents the SMTP server from rejecting mail permanently
# (by changing 5xx replies into 4xx replies). However, soft_bounce
# is no cure for address rewriting mistakes or mail routing mistakes.
#
#soft_bounce = no

# LOCAL PATHNAME INFORMATION
#
# The queue_directory specifies the location of the Postfix queue.
# This is also the root directory of Postfix daemons that run chrooted.
# See the files in examples/chroot-setup for setting up Postfix chroot
# environments on different UNIX systems.
#
queue_directory = /var/spool/postfix

# The command_directory parameter specifies the location of all
# postXXX commands.
#
command_directory = /usr/sbin

# The daemon_directory parameter specifies the location of all Postfix
# daemon programs (i.e. programs listed in the master.cf file). This
# directory must be owned by root.
#
daemon_directory = /usr/libexec/postfix

# QUEUE AND PROCESS OWNERSHIP
#
# The mail_owner parameter specifies the owner of the Postfix queue
# and of most Postfix daemon processes.  Specify the name of a user
# account THAT DOES NOT SHARE ITS USER OR GROUP ID WITH OTHER ACCOUNTS
# AND THAT OWNS NO OTHER FILES OR PROCESSES ON THE SYSTEM.  In
# particular, don't specify nobody or daemon. PLEASE USE A DEDICATED
# USER.
#
mail_owner = postfix

# The default_privs parameter specifies the default rights used by
# the local delivery agent for delivery to external file or command.
# These rights are used in the absence of a recipient user context.
```

```
# DO NOT SPECIFY A PRIVILEGED USER OR THE POSTFIX OWNER.
#
#default_privs = nobody

# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
#myhostname = host.domain.tld
#myhostname = virtual.domain.tld

myhostname = mail.labreti4.it

# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
# $mydomain is used as a default value for many other configuration
# parameters.
#
#mydomain = domain.tld

mydomain = labreti4.it

# SENDING MAIL
#
# The myorigin parameter specifies the domain that locally-posted
# mail appears to come from. The default is to append $myhostname,
# which is fine for small sites.  If you run a domain with multiple
# machines, you should (1) change this to $mydomain and (2) set up
# a domain-wide alias database that aliases each user to
# user@that.users.mailhost.
#
# For the sake of consistency between sender and recipient addresses,
# myorigin also specifies the default domain name that is appended
# to recipient addresses that have no @domain part.
#

myorigin = $mydomain

# RECEIVING MAIL

# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on.  By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost
#inet_interfaces = localhost

inet_interfaces = all


# The proxy_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on by way of a
# proxy or network address translation unit. This setting extends
```

```
# the address list specified with the inet_interfaces parameter.
#
# You must specify your proxy/NAT addresses when your system is a
# backup MX host for other domains, otherwise mail delivery loops
# will happen when the primary MX host is down.
#
#proxy_interfaces =
#proxy_interfaces = 1.2.3.4

# The mydestination parameter specifies the list of domains that this
# machine considers itself the final destination for.
#
# These domains are routed to the delivery agent specified with the
# local_transport parameter setting. By default, that is the UNIX
# compatible delivery agent that lookups all recipients in /etc/passwd
# and /etc/aliases or their equivalent.
#
# The default is $myhostname + localhost.$mydomain.  On a mail domain
# gateway, you should also include $mydomain.
#
# Do not specify the names of virtual domains - those domains are
# specified elsewhere (see VIRTUAL_README).
#
# Do not specify the names of domains that this machine is backup MX
# host for. Specify those names via the relay_domains settings for
# the SMTP server, or use permit_mx_backup if you are lazy (see
# STANDARD_CONFIGURATION_README).
#
# The local machine is always the final destination for mail addressed
# to user@[the.net.work.address] of an interface that the mail system
# receives mail on (see the inet_interfaces parameter).
#
# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
#mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$mydomain, www.$mydomain, ftp.$mydomain


mydestination = $myhostname, localhost.$mydomain, $mydomain

# REJECTING MAIL FOR UNKNOWN LOCAL USERS
#
# The local_recipient_maps parameter specifies optional lookup tables
# with all names or addresses of users that are local with respect
# to $mydestination, $inet_interfaces or $proxy_interfaces.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown local users. This parameter is defined by default.
#
# To turn off local recipient checking in the SMTP server, specify
# local_recipient_maps = (i.e. empty).
#
# The default setting assumes that you use the default Postfix local
# delivery agent for local delivery. You need to update the
# local_recipient_maps setting if:
```

```
#
# - You define $mydestination domain recipients in files other than
#   /etc/passwd, /etc/aliases, or the $virtual_alias_maps files.
#   For example, you define $mydestination domain recipients in
#   the $virtual_mailbox_maps files.
#
# - You redefine the local delivery agent in master.cf.
#
# - You redefine the "local_transport" setting in main.cf.
#
# - You use the "luser_relay", "mailbox_transport", or "fallback_transport"
#   feature of the Postfix local delivery agent (see local(8)).
#
# Details are described in the LOCAL_RECIPIENT_README file.
#
# Beware: if the Postfix SMTP server runs chrooted, you probably have
# to access the passwd file via the proxymap service, in order to
# overcome chroot restrictions. The alternative, having a copy of
# the system passwd file in the chroot jail is just not practical.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify a bare username, an @domain.tld
# wild-card, or specify a user@domain.tld address.
#
#local_recipient_maps = unix:passwd.byname $alias_maps
#local_recipient_maps = proxy:unix:passwd.byname $alias_maps
#local_recipient_maps =

# The unknown_local_recipient_reject_code specifies the SMTP server
# response code when a recipient domain matches $mydestination or
# ${proxy,inet}_interfaces, while $local_recipient_maps is non-empty
# and the recipient address or address local-part is not found.
#
# The default setting is 550 (reject mail) but it is safer to start
# with 450 (try again later) until you are certain that your
# local_recipient_maps settings are OK.
#
unknown_local_recipient_reject_code = 550

# TRUST AND RELAY CONTROL

# The mynetworks parameter specifies the list of "trusted" SMTP
# clients that have more privileges than "strangers".
#
# In particular, "trusted" SMTP clients are allowed to relay mail
# through Postfix.  See the smtpd_recipient_restrictions parameter
# in postconf(5).
#
# You can specify the list of "trusted" network addresses by hand
# or you can let Postfix do it for you (which is the default).
#
# By default (mynetworks_style = subnet), Postfix "trusts" SMTP
# clients in the same IP subnetworks as the local machine.
# On Linux, this does works correctly only with interfaces specified
# with the "ifconfig" command.
#
# Specify "mynetworks_style = class" when Postfix should "trust" SMTP
# clients in the same IP class A/B/C networks as the local machine.
# Don't do this with a dialup site - it would cause Postfix to "trust"
# your entire provider's network.  Instead, specify an explicit
# mynetworks list by hand, as described below.
#
# Specify "mynetworks_style = host" when Postfix should "trust"
```

```
# only the local machine.
#
#mynetworks_style = subnet
#mynetworks_style = host

mynetworks_style = class

# Alternatively, you can specify the mynetworks list by hand, in
# which case Postfix ignores the mynetworks_style setting.
#
# Specify an explicit list of network/netmask patterns, where the
# mask specifies the number of bits in the network part of a host
# address.
#
# You can also specify the absolute pathname of a pattern file instead
# of listing the patterns here. Specify type:table for table-based lookups
# (the value on the table right-hand side is not used).
#
#mynetworks = 168.100.189.0/28, 127.0.0.0/8
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table
mynetworks = 192.168.0.0/24, 192.168.1.0/24, 127.0.0.0/8

# The relay_domains parameter restricts what destinations this system will
# relay mail to.  See the smtpd_recipient_restrictions description in
# postconf(5) for detailed information.
#
# By default, Postfix relays mail
# - from "trusted" clients (IP address matches $mynetworks) to any destination,
# - from "untrusted" clients to destinations that match $relay_domains or
#   subdomains thereof, except addresses with sender-specified routing.
# The default relay_domains value is $mydestination.
#
# In addition to the above, the Postfix SMTP server by default accepts mail
# that Postfix is final destination for:
# - destinations that match $inet_interfaces or $proxy_interfaces,
# - destinations that match $mydestination
# - destinations that match $virtual_alias_domains,
# - destinations that match $virtual_mailbox_domains.
# These destinations do not need to be listed in $relay_domains.
#
# Specify a list of hosts or domains, /file/name patterns or type:name
# lookup tables, separated by commas and/or whitespace.  Continue
# long lines by starting the next line with whitespace. A file name
# is replaced by its contents; a type:name table is matched when a
# (parent) domain appears as lookup key.
#
# NOTE: Postfix will not automatically forward mail for domains that
# list this system as their primary or backup MX host. See the
# permit_mx_backup restriction description in postconf(5).
#
#relay_domains = $mydestination

# INTERNET OR INTRANET

# The relayhost parameter specifies the default host to send mail to
# when no entry is matched in the optional transport(5) table. When
# no relayhost is given, mail is routed directly to the destination.
#
# On an intranet, specify the organizational domain name. If your
# internal DNS uses no MX records, specify the name of the intranet
# gateway host instead.
#
```

```
# In the case of SMTP, specify a domain, host, host:port, [host]:port,
# [address] or [address]:port; the form [host] turns off MX lookups.
#
# If you're connected via UUCP, see also the default_transport parameter.
#
#relayhost = $mydomain
#relayhost = [gateway.my.domain]
#relayhost = [mailserver.isp.tld]
#relayhost = uucphost
#relayhost = [an.ip.add.ress]

# REJECTING UNKNOWN RELAY USERS
#
# The relay_recipient_maps parameter specifies optional lookup tables
# with all addresses in the domains that match $relay_domains.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown relay users. This feature is off by default.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify an @domain.tld wild-card, or specify
# a user@domain.tld address.
#
#relay_recipient_maps = hash:/etc/postfix/relay_recipients
#
#smtpd_helo_restrictions =
#check_client_access = tipo_file:percorso_file

smtpd_client_restrictions = hash:/etc/postfix/host-reject
smtpd_sender_restrictions = hash:/etc/postfix/address-reject
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination


# INPUT RATE CONTROL
#
# The in_flow_delay configuration parameter implements mail input
# flow control. This feature is turned on by default, although it
# still needs further development (it's disabled on SCO UNIX due
# to an SCO bug).
#
# A Postfix process will pause for $in_flow_delay seconds before
# accepting a new message, when the message arrival rate exceeds the
# message delivery rate. With the default 100 SMTP server process
# limit, this limits the mail inflow to 100 messages a second more
# than the number of messages delivered per second.
#
# Specify 0 to disable the feature. Valid delays are 0..10.
#
#in_flow_delay = 1s

# ADDRESS REWRITING
#
# The ADDRESS_REWRITING_README document gives information about
# address masquerading or other forms of address rewriting including
# username->Firstname.Lastname mapping.

# ADDRESS REDIRECTION (VIRTUAL DOMAIN)
#
# The VIRTUAL_README document gives information about the many forms
# of domain hosting that Postfix supports.
virtual_maps = hash:/etc/postfix/virtual

# "USER HAS MOVED" BOUNCE MESSAGES
```

```
#
# See the discussion in the ADDRESS_REWRITING_README document.

# TRANSPORT MAP
#
# See the discussion in the ADDRESS_REWRITING_README document.

# ALIAS DATABASE
#
# The alias_maps parameter specifies the list of alias databases used
# by the local delivery agent. The default list is system dependent.
#
# On systems with NIS, the default is to search the local alias
# database, then the NIS alias database. See aliases(5) for syntax
# details.
#
# If you change the alias database, run "postalias /etc/aliases" (or
# wherever your system stores the mail alias file), or simply run
# "newaliases" to build the necessary DBM or DB file.
#
# It will take a minute or so before changes become visible.  Use
# "postfix reload" to eliminate the delay.
#
#alias_maps = dbm:/etc/aliases
#alias_maps = hash:/etc/aliases, nis:mail.aliases
#alias_maps = netinfo:/aliases

alias_maps = hash:/etc/aliases


# The alias_database parameter specifies the alias database(s) that
# are built with "newaliases" or "sendmail -bi".  This is a separate
# configuration parameter, because alias_maps (see above) may specify
# tables that are not necessarily all under control by Postfix.
#
#alias_database = dbm:/etc/aliases
#alias_database = dbm:/etc/mail/aliases
#alias_database = hash:/etc/aliases, hash:/opt/majordomo/aliases

alias_database = hash:/etc/aliases

# ADDRESS EXTENSIONS (e.g., user+foo)
#
# The recipient_delimiter parameter specifies the separator between
# user names and address extensions (user+foo). See canonical(5),
# local(8), relocated(5) and virtual(5) for the effects this has on
# aliases, canonical, virtual, relocated and .forward file lookups.
# Basically, the software tries user+foo and .forward+foo before
# trying user and .forward.
#
#recipient_delimiter = +

# DELIVERY TO MAILBOX
#
# The home_mailbox parameter specifies the optional pathname of a
# mailbox file relative to a user's home directory. The default
# mailbox file is /var/spool/mail/user or /var/mail/user.  Specify
# "Maildir/" for qmail-style delivery (the / is required).
#
#home_mailbox = Mailbox
#home_mailbox = Maildir/

# The mail_spool_directory parameter specifies the directory where
```

```
# UNIX-style mailboxes are kept. The default setting depends on the
# system type.
#
#mail_spool_directory = /var/mail
#mail_spool_directory = /var/spool/mail

# The mailbox_command parameter specifies the optional external
# command to use instead of mailbox delivery. The command is run as
# the recipient with proper HOME, SHELL and LOGNAME environment settings.
# Exception:  delivery for root is done as $default_user.
#
# Other environment variables of interest: USER (recipient username),
# EXTENSION (address extension), DOMAIN (domain part of address),
# and LOCAL (the address localpart).
#
# Unlike other Postfix configuration parameters, the mailbox_command
# parameter is not subjected to $parameter substitutions. This is to
# make it easier to specify shell syntax (see example below).
#
          # Avoid shell meta characters because they will force Postfix to run
# an expensive shell process. Procmail alone is expensive enough.
#
# IF YOU USE THIS TO DELIVER MAIL SYSTEM-WIDE, YOU MUST SET UP AN
# ALIAS THAT FORWARDS MAIL FOR ROOT TO A REAL USER.
#
#mailbox_command = /some/where/procmail
#mailbox_command = /some/where/procmail -a "$EXTENSION"

mailbox_command = /usr/bin/procmail

# The mailbox_transport specifies the optional transport in master.cf
# to use after processing aliases and .forward files. This parameter
# has precedence over the mailbox_command, fallback_transport and
# luser_relay parameters.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
        # file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#mailbox_transport = lmtp:unix:/file/name
#mailbox_transport = cyrus

# The fallback_transport specifies the optional transport in master.cf
# to use for recipients that are not found in the UNIX passwd database.
# This parameter has precedence over the luser_relay parameter.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#fallback_transport = lmtp:unix:/file/name
```

```
#fallback_transport = cyrus
#fallback_transport =

# The luser_relay parameter specifies an optional destination address
# for unknown recipients.  By default, mail for unknown@$mydestination,
# unknown@[$inet_interfaces] or unknown@[$proxy_interfaces] is returned
# as undeliverable.
#
# The following expansions are done on luser_relay: $user (recipient
# username), $shell (recipient shell), $home (recipient home directory),
# $recipient (full recipient address), $extension (recipient address
# extension), $domain (recipient domain), $local (entire recipient
# localpart), $recipient_delimiter. Specify ${name?value} or
# ${name:value} to expand value only when $name does (does not) exist.
#
# luser_relay works only for the default Postfix local delivery agent.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must specify "local_recipient_maps =" (i.e. empty) in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#luser_relay = $user@other.host
#luser_relay = $local@other.host
#luser_relay = admin+$local

# JUNK MAIL CONTROLS
#
# The controls listed here are only a very small subset. The file
# SMTPD_ACCESS_README provides an overview.

# The header_checks parameter specifies an optional table with patterns
# that each logical message header is matched against, including
# headers that span multiple physical lines.
#
# By default, these patterns also apply to MIME headers and to the
# headers of attached messages. With older Postfix versions, MIME and
# attached message headers were treated as body text.
#
# For details, see "man header_checks".
#
#header_checks = regexp:/etc/postfix/header_checks

# FAST ETRN SERVICE
#
# Postfix maintains per-destination logfiles with information about
# deferred mail, so that mail can be flushed quickly with the SMTP
# "ETRN domain.tld" command, or by executing "sendmail -qRdomain.tld".
# See the ETRN_README document for a detailed description.
#
# The fast_flush_domains parameter controls what destinations are
# eligible for this service. By default, they are all domains that
# this server is willing to relay mail to.
#
#fast_flush_domains = $relay_domains

# SHOW SOFTWARE VERSION OR NOT
#
# The smtpd_banner parameter specifies the text that follows the 220
# code in the SMTP server's greeting banner. Some people like to see
# the mail version advertised. By default, Postfix shows no version.
#
 You MUST specify $myhostname at the start of the text. That is an
```

```
# RFC requirement. Postfix itself does not care.
#
#smtpd_banner = $myhostname ESMTP $mail_name
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)

# PARALLEL DELIVERY TO THE SAME DESTINATION
#
# How many parallel deliveries to the same user or domain? With local
# delivery, it does not make sense to do massively parallel delivery
# to the same user, because mailbox updates must happen sequentially,
# and expensive pipelines in .forward files can cause disasters when
# too many are run at the same time. With SMTP deliveries, 10
# simultaneous connections to the same domain could be sufficient to
# raise eyebrows.
#
# Each message delivery transport has its XXX_destination_concurrency_limit
# parameter.  The default is $default_destination_concurrency_limit for
# most delivery transports. For the local delivery agent the default is 2.

local_destination_concurrency_limit = 2
default_destination_concurrency_limit = 20
initial_destination_concurrency = 2
default_process_limit = 100
smtpd_error_sleep_time = 1
smtpd_soft_error_limit = 10
smtpd_hard_error_limit = 20

# DEBUGGING CONTROL
#
# The debug_peer_level parameter specifies the increment in verbose
# logging level when an SMTP client or server host name or address
# matches a pattern in the debug_peer_list parameter.
#
debug_peer_level = 2

# The debug_peer_list parameter specifies an optional list of domain
# or network patterns, /file/name patterns or type:name tables. When
# an SMTP client or server host name or address matches a pattern,
# increase the verbose logging level by the amount specified in the
# debug_peer_level parameter.
#
#debug_peer_list = 127.0.0.1
#debug_peer_list = some.domain

# The debugger_command specifies the external command that is executed
# when a Postfix daemon program is run with the -D option.
#
# Use "command .. & sleep 5" so that the debugger can attach before
# the process marches on. If you use an X-based debugger, be sure to
# set up your XAUTHORITY environment variable before starting Postfix.
#
debugger_command =
         PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
         xxgdb $daemon_directory/$process_name $process_id & sleep 5

# If you don't have X installed on the Postfix machine, try:
# debugger_command =
#       PATH=/bin:/usr/bin:/usr/local/bin; export PATH; (echo cont;
#       echo where) | gdb $daemon_directory/$process_name $process_id 2>&1
#       >$config_directory/$process_name.$process_id.log & sleep 5

# INSTALL-TIME CONFIGURATION INFORMATION
#
```

```
# The following parameters are used when installing a new Postfix version.
#
# sendmail_path: The full pathname of the Postfix sendmail command.
# This is the Sendmail-compatible mail posting interface.
#
sendmail_path = /usr/sbin/sendmail.postfix

# newaliases_path: The full pathname of the Postfix newaliases command.
# This is the Sendmail-compatible command to build alias databases.
#
newaliases_path = /usr/bin/newaliases.postfix

# mailq_path: The full pathname of the Postfix mailq command.  This
# is the Sendmail-compatible mail queue listing command.
#
mailq_path = /usr/bin/mailq.postfix

# setgid_group: The group for mail submission and queue management
# commands.  This must be a group name with a numerical group ID that
# is not shared with other accounts, not even with the Postfix account.
#
setgid_group = postdrop

# html_directory: The location of the Postfix HTML documentation.
#
html_directory = no

# manpage_directory: The location of the Postfix on-line manual pages.
#
manpage_directory = /usr/share/man

# sample_directory: The location of the Postfix sample configuration files.
# This parameter is obsolete as of Postfix 2.1.
#
sample_directory = /usr/share/doc/postfix-2.1.5/samples

# readme_directory: The location of the Postfix README files.
#
readme_directory = /usr/share/doc/postfix-2.1.5/README_FILES
```

File di configurazione di Postfix `/etc/postfix/master.cf`

```
# Postfix master process configuration file.  Each logical line
# describes how a Postfix daemon program should be run.
#
# A logical line starts with non-whitespace, non-comment text.
# Empty lines and whitespace-only lines are ignored, as are comment
# lines whose first non-whitespace character is a `#'.
# A line that starts with whitespace continues a logical line.
#
# The fields that make up each line are described below. A "-" field
# value requests that a default value be used for that field.
#
# Service: any name that is valid for the specified transport type
# (the next field).  With INET transports, a service is specified as
# host:port.  The host part (and colon) may be omitted. Either host
# or port may be given in symbolic form or in numeric form. Examples
# for the SMTP server:  localhost:smtp receives mail via the loopback
# interface only; 10025 receives mail on port 10025.
#
```

```
# Transport type: "inet" for Internet sockets, "unix" for UNIX-domain
# sockets, "fifo" for named pipes.
#
# Private: whether or not access is restricted to the mail system.
# Default is private service.  Internet (inet) sockets can't be private.
#
# Unprivileged: whether the service runs with root privileges or as
# the owner of the Postfix system (the owner name is controlled by the
# mail_owner configuration variable in the main.cf file). Only the
# pipe, virtual and local delivery daemons require privileges.
#
# Chroot: whether or not the service runs chrooted to the mail queue
# directory (pathname is controlled by the queue_directory configuration
# variable in the main.cf file). Presently, all Postfix daemons can run
# chrooted, except for the pipe, virtual and local delivery daemons.
# The proxymap server can run chrooted, but doing so defeats most of
# the purpose of having that service in the first place.
# The files in the examples/chroot-setup subdirectory describe how
# to set up a Postfix chroot environment for your type of machine.
#
# Wakeup time: automatically wake up the named service after the
# specified number of seconds. A ? at the end of the wakeup time
# field requests that wake up events be sent only to services that
# are actually being used.  Specify 0 for no wakeup. Presently, only
# the pickup, queue manager and flush daemons need a wakeup timer.
#
# Max procs: the maximum number of processes that may execute this
# service simultaneously. Default is to use a globally configurable
# limit (the default_process_limit configuration parameter in main.cf).
# Specify 0 for no process count limit.
#
# Command + args: the command to be executed. The command name is
# relative to the Postfix program directory (pathname is controlled by
# the daemon_directory configuration variable). Adding one or more
# -v options turns on verbose logging for that service; adding a -D
# option enables symbolic debugging (see the debugger_command variable
# in the main.cf configuration file). See individual command man pages
# for specific command-line options, if any.
#
# General main.cf options can be overridden for specific services.
# To override one or more main.cf options, specify them as arguments
# below, preceding each option by "-o".  There must be no whitespace
# in the option itself (separate multiple values for an option by
# commas).
#
# In order to use the "uucp" message tranport below, set up entries
# in the transport table.
#
# In order to use the "cyrus" message transport below, configure it
# in main.cf as the mailbox_transport.
#
# SPECIFY ONLY PROGRAMS THAT ARE WRITTEN TO RUN AS POSTFIX DAEMONS.
# ALL DAEMONS SPECIFIED HERE MUST SPEAK A POSTFIX-INTERNAL PROTOCOL.
#
# DO NOT SHARE THE POSTFIX QUEUE BETWEEN MULTIPLE POSTFIX INSTANCES.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd
#smtps    inet  n       -       n       -       -       smtpd
#  -o smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes
```

```
#submission   inet    n       -       n       -       -       smtpd
#  -o smtpd_enforce_tls=yes -o smtpd_sasl_auth_enable=yes -o
smtpd_etrn_restrictions=reject
#628      inet  n       -       n       -       -       qmqpd
pickup    fifo  n       -       n       60      1       pickup
cleanup   unix  n       -       n       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
#qmgr     fifo  n       -       n       300     1       oqmgr
#tlsmgr   fifo  -       -       n       300     1       tlsmgr
rewrite   unix  -       -       n       -       -       trivial-rewrite
bounce    unix  -       -       n       -       0       bounce
defer     unix  -       -       n       -       0       bounce
trace     unix  -       -       n       -       0       bounce
verify    unix  -       -       n       -       1       verify
flush     unix  n       -       n       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
smtp      unix  -       -       n       -       -       smtp
relay     unix  -       -       n       -       -       smtp
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq     unix  n       -       n       -       -       showq
error     unix  -       -       n       -       -       error
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       n       -       -       lmtp
anvil     unix  -       -       n       -       1       anvil
#
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# maildrop. See the Postfix MAILDROP_README file for details.
#
maildrop  unix  -       n       n       -       -       pipe
  flags=DRhu user=vmail argv=/usr/local/bin/maildrop -d ${recipient}
#
# The Cyrus deliver program has changed incompatibly, multiple times.
#
old-cyrus unix  -       n       n       -       -       pipe
  flags=R user=cyrus argv=/usr/lib/cyrus-imapd/deliver -e -m ${extension}
${user}
# Cyrus 2.1.5 (Amos Gouaux)
# Also specify in main.cf: cyrus_destination_recipient_limit=1
cyrus     unix  -       n       n       -       -       pipe
  user=cyrus argv=/usr/lib/cyrus-imapd/deliver -e -r ${sender} -m ${extension}
${user}
uucp      unix  -       n       n       -       -       pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
ifmail    unix  -       n       n       -       -       pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp     unix  -       n       n       -       -       pipe
  flags=Fq. user=foo argv=/usr/local/sbin/bsmtp -f $sender $nexthop $recipient
```

File `/etc/postfix/aliases`

```
#
#  Aliases in this file will NOT be expanded in the header from
#  Mail, but WILL be visible over networks or from /bin/mail.
#
#       >>>>>>>>>>      The program "newaliases" must be run after
#       >> NOTE >>      this file is updated for any changes to
#       >>>>>>>>>>      show through to sendmail.
#
```

```
# Basic system aliases -- these MUST be present.
mailer-daemon:  postmaster
postmaster:     root

# General redirections for pseudo accounts.
bin:            root
daemon:         root
adm:            root
lp:             root
sync:           root
shutdown:       root
halt:           root
mail:           root
news:           root
uucp:           root
operator:       root
games:          root
gopher:         root
ftp:            root
nobody:         root
radiusd:        root
nut:            root
dbus:           root
vcsa:           root
canna:          root
wnn:            root
rpm:            root
nscd:           root
pcap:           root
apache:         root
webalizer:      root
dovecot:        root
fax:            root
quagga:         root
radvd:          root
pvm:            root
amanda:         root
privoxy:        root
ident:          root
named:          root
xfs:            root
gdm:            root
mailnull:       root
postgres:       root
sshd:           root
smmsp:          root
postfix:        root
netdump:        root
ldap:           root
squid:          root
ntp:            root
mysql:          root
desktop:        root
rpcuser:        root
rpc:            root
nfsnobody:      root

ingres:         root
system:         root
toor:           root
manager:        root
dumper:         root
abuse:          root
```

```
# mailman aliases
mailman:        postmaster
mailman-owner:  mailman

newsadm:        news
newsadmin:      news
usenet:         news
ftpadm:         ftp
ftpadmin:       ftp
ftp-adm:        ftp
ftp-admin:      ftp
www:            webmaster
webmaster:      root
noc:            root
security:       root
hostmaster:     root
info:           postmaster
marketing:      postmaster
sales:          postmaster
support:        postmaster


# trap decode to catch security attacks
decode:         root

# Person who should get root's mail
root:           amministratore
```

File /etc/postfix/virtual

```
# VIRTUAL(5)                                                VIRTUAL(5)
#
# NAME
#       virtual - format of Postfix virtual alias table
#
# SYNOPSIS
#       postmap /etc/postfix/virtual
#
#       postmap -q "string" /etc/postfix/virtual
#
#       postmap -q - /etc/postfix/virtual <inputfile
#
# DESCRIPTION
#       The  optional virtual alias table specifies address alias-
#       ing for arbitrary local or non-local recipient  addresses.
#       Virtual  aliasing is recursive, and is done by the Postfix
#       cleanup(8) daemon.
#
#       The main applications of virtual aliasing are:
#
#       o      To redirect mail for one address  to  one  or  more
#              addresses.
#
  #
#       o      To   implement  virtual  alias  domains  where  all
#              addresses  are  aliased  to  addresses   in   other
#              domains.
#
#              Virtual  alias  domains are not to be confused with
```

```
#               the virtual mailbox domains  that  are  implemented
#               with  the  Postfix  virtual(8) mail delivery agent.
#               With  virtual  mailbox  domains,   each   recipient
#               address can have its own mailbox.
#
#       Virtual  aliasing  is  applied  only to recipient envelope
#       addresses, and does not  affect  message  headers.   Think
#       Sendmail  rule  set S0, if you like. Use canonical(5) map-
#       ping to rewrite header and envelope addresses in  general.
#
#       Normally,  the  virtual alias table is specified as a text
#       file that serves as input to the postmap(1) command.   The
#       result,  an  indexed file in dbm or db format, is used for
#       fast searching by the mail  system.  Execute  the  command
#       postmap  /etc/postfix/virtual  in  order  to  rebuild  the
#       indexed file after changing the text file.
 #
#       When the table is provided via other means  such  as  NIS,
#       LDAP  or  SQL,  the  same lookups are done as for ordinary
#       indexed files.
#
#       Alternatively, the table can be  provided  as  a  regular-
#       expression map where patterns are given as regular expres-
#       sions, or lookups can be directed to TCP-based server.  In
#       that  case,  the  lookups are done in a slightly different
#       way as described below under "REGULAR  EXPRESSION  TABLES"
#       and "TCP-BASED TABLES".
#
# TABLE FORMAT
#       The input format for the postmap(1) command is as follows:
#
#       pattern result
#              When pattern matches a mail address, replace it  by
#              the corresponding result.
#
#       blank lines and comments
#              Empty  lines and whitespace-only lines are ignored,
                as are lines whose first  non-whitespace  character
#              is a `#'.
#
#       multi-line text
#              A  logical  line starts with non-whitespace text. A
#              line that starts with whitespace continues a  logi-
#              cal line.
#
#       With lookups from indexed files such as DB or DBM, or from
#       networked tables such as NIS, LDAP or  SQL,  patterns  are
#       tried in the order as listed below:
#
#       user@domain address, address, ...
#              Mail  for  user@domain  is  redirected  to address.
#              This form has the highest precedence.
#
#       user address, address, ...
#              Mail for user@site is redirected  to  address  when
#              site  is equal to $myorigin, when site is listed in
#              $mydestination,   or   when   it   is   listed   in
#              $inet_interfaces or $proxy_interfaces.
#
#              This  functionality  overlaps with functionality of
#              the local aliases(5) database.  The  difference  is
#              that  virtual  mapping  can be applied to non-local
#              addresses.
```

```
#
#       @domain address, address, ...
#              Mail for any user in domain is redirected to
#              address.  This form has the lowest precedence.
#
#       In  all the above forms, when address has the form @other-
#       domain, the result is the same user in otherdomain.   This
#       works for the first address in the expansion only.
#
# ADDRESS EXTENSION
#       When a mail address localpart contains the optional recip-
#       ient delimiter (e.g., user+foo@domain), the  lookup  order
#       becomes: user+foo@domain, user@domain, user+foo, user, and
#       @domain.
#
#       The  propagate_unmatched_extensions   parameter   controls
#       whether  an  unmatched  address extension (+foo) is propa-
#       gated to the result of table lookup.
#
# VIRTUAL ALIAS DOMAINS
#       Besides virtual aliases, the virtual alias table can  also
#       be used to implement virtual alias domains. With a virtual
#       alias domain,  all  recipient  addresses  are  aliased  to
#       addresses in other domains.
#
#       Virtual alias domains are not to be confused with the vir-
#       tual mailbox domains that are implemented with the Postfix
#       virtual(8)  mail  delivery  agent.  With  virtual  mailbox
#       domains, each recipient address can have its own  mailbox.
#
#       With  a  virtual  alias domain, the virtual domain has its
#       own user name space. Local  (i.e.  non-virtual)  usernames
#       are  not visible in a virtual alias domain. In particular,
#       local aliases(5) and local mailing lists are  not  visible
#       as localname@virtual-alias.domain.
#
#       Support for a virtual alias domain looks like:
#
#       /etc/postfix/main.cf:
#           virtual_alias_maps = hash:/etc/postfix/virtual
#
#           Note:  some systems use dbm databases instead of hash.
#           See the output from postconf -m for available database
#           types.
#
#       /etc/postfix/virtual:
#           virtual-alias.domain anything (right-hand content does not matter)
#           postmaster@virtual-alias.domain      postmaster
#           user1@virtual-alias.domain   address1
#           user2@virtual-alias.domain   address2, address3
#
#       The  virtual-alias.domain anything entry is required for a
#       virtual alias domain. Without this entry, mail is rejected
#       with  "relay  access  denied", or bounces with "mail loops
#       back to myself".
#
#       Do not specify virtual alias domain names in  the  main.cf
#       mydestination or relay_domains configuration parameters.
#
#       With  a  virtual  alias  domain,  the  Postfix SMTP server
#       accepts  mail  for  known-user@virtual-alias.domain,   and
#       rejects   mail   for   unknown-user@virtual-alias.domain  as
#       undeliverable.
```

```
#
#       Instead of specifying the virtual alias  domain  name  via
#       the  virtual_alias_maps table, you may also specify it via
#       the main.cf virtual_alias_domains configuration parameter.
#       This  latter parameter uses the same syntax as the main.cf
#       mydestination configuration parameter.
#
# REGULAR EXPRESSION TABLES
#       This section describes how the table lookups  change  when
#       the table is given in the form of regular expressions. For
#       a description of regular expression lookup  table  syntax,
#       see regexp_table(5) or pcre_table(5).
#
#       Each  pattern  is  a regular expression that is applied to
#       the entire address being looked up. Thus, user@domain mail
#       addresses  are  not  broken up into their user and @domain
#       constituent parts, nor is user+foo broken up into user and
#       foo.
#
#       Patterns  are  applied  in  the  order as specified in the
#       table, until a pattern is found that  matches  the  search
#       string.
#
#       Results  are  the  same as with indexed file lookups, with
#       the additional feature that parenthesized substrings  from
#       the pattern can be interpolated as $1, $2 and so on.
#
# TCP-BASED TABLES
#       This  section  describes how the table lookups change when
#       lookups are directed to a TCP-based server. For a descrip-
#       tion   of   the   TCP   client/server   lookup   protocol,   see
#       tcp_table(5).  This feature is not  available  in  Postfix
#       version 2.1.
#
#       Each lookup operation uses the entire address once.  Thus,
#       user@domain mail addresses are not broken  up  into  their
#       user and @domain constituent parts, nor is user+foo broken
#       up into user and foo.
#
#       Results are the same as with indexed file lookups.
#
# BUGS
#       The table format does not understand quoting  conventions.
#
# CONFIGURATION PARAMETERS
#       The  following  main.cf parameters are especially relevant
#       to this topic. See the Postfix  main.cf  file  for  syntax
#       details  and  for  default  values. Use the postfix reload
#       command after a configuration change.
#
#       virtual_alias_maps
#              List of virtual aliasing tables.
#
#       virtual_alias_domains
#              List of virtual alias domains. This uses  the  same
#              syntax as the mydestination parameter.
#
#       propagate_unmatched_extensions
#              A  list  of  address rewriting or forwarding mecha-
#              nisms that propagate an address extension from  the
#              original  address  to  the result.  Specify zero or
#              more of  canonical,  virtual,  alias,  forward,  or
#              include.
```

```
#
#       Other parameters of interest:
#
#       inet_interfaces
#              The  network  interface  addresses that this system
#              receives mail on.  You need to stop and start Post-
#              fix when this parameter changes.
#
#       mydestination
#              List  of  domains  that  this mail system considers
#              local.
#
#       myorigin
#              The domain that is appended  to  any  address  that
#              does not have a domain.
#
#       owner_request_special
#              Give special treatment to owner-xxx and xxx-request
#              addresses.
#
#       proxy_interfaces
#              Other interfaces that this machine receives mail on
#              by way of a proxy agent or network address transla-
#              tor.
#
# SEE ALSO
#       cleanup(8), canonicalize and enqueue mail
#       postmap(1), Postfix lookup table manager
#       postconf(5), configuration parameters
#       canonical(5), canonical address mapping
#
# README FILES
#       Use "postconf readme_directory" or  "postconf  html_direc-
#       tory" to locate this information.
#       DATABASE_README, Postfix lookup table overview
#       ADDRESS_REWRITING_README, address rewriting guide
#       VIRTUAL_README, domain hosting guide
#
# LICENSE
#       The  Secure  Mailer  license must be distributed with this
#       software.
#
# AUTHOR(S)
#       Wietse Venema
#       IBM T.J. Watson Research
#       P.O. Box 704
#       Yorktown Heights, NY 10598, USA
#
labreti4.it             anything
postmaster@labreti4.it  admin
postino@labreti4.it     postino
spam@labreti4.it        spam
@labreti4.it            amministratore@labreti4.it
                                                      VIRTUAL(5)
```

Procmailrc

File di configurazione di Procmail `/etc/procmailrc`

```
# SpamAssassin sample procmailrc
# ==============================

# The following line is only used if you use a system-wide /etc/procmailrc.
# See procmailrc(5) for infos on what it exactly does, the short version:
```

```
#  * It ensures that the correct user is passed to spamd if spamc is used
#  * The folders the mail is filed to later on is owned by the user, not
#    root.

DROPPRIVS=yes
LOGFILE=/var/log/procmail.log


# Pipe the mail through spamassassin (replace 'spamassassin' with 'spamc'
# if you use the spamc/spamd combination)
#
# The condition line ensures that only messages smaller than 250 kB
# (250 * 1024 = 256000 bytes) are processed by SpamAssassin. Most spam
# isn't bigger than a few k and working with big messages can bring
# SpamAssassin to its knees.
#
# The lock file ensures that only 1 spamassassin invocation happens
# at 1 time, to keep the load down.
#
:0fw: spamassassin.lock
* < 256000
| /usr/bin/spamassassin

# Mails with a score of 15 or higher are almost certainly spam (with 0.05%
# false positives according to rules/STATISTICS.txt). Let's put them in a
# different mbox. (This one is optional.)
:0:
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
almost-certainly-spam

# All mail tagged as spam (eg. with a score higher than the set threshold)
# is moved to "probably-spam".
:0:
* ^X-Spam-Status: Yes
probably-spam
#Se trova il flag nell'header dell'email (inserito da spamassassin per le email
ritenute spam) sposta la mail in una mailbox chiamata appunto "spam"
:0:/tmp/spam.lock
*^X-Spam-Flag: YES
/var/spool/mail/spam

# Work around procmail bug: any output on stderr will cause the "F" in "From"
# to be dropped.  This will re-add it.
# NOTE: This is probably NOT needed in recent versions of procmail
:0
* ^^rom[ ]
{
  LOG="*** Dropped F off From_ header! Fixing up. "

  :0 fhw
  | sed -e '1s/^/F/'
}
```

Dovecot

File di configurazione di Procmail /etc/dovecot.conf

```
## Dovecot 1.0 configuration file

# Base directory where to store runtime data.
base_dir = /var/run/dovecot/
```

```
# Protocols we want to be serving:
#  imap imaps pop3 pop3s
protocols = imap pop3

# IP or host address where to listen in for connections. It's not currently
# possible to specify multiple addresses. "*" listens in all IPv4 interfaces.
# "[::]" listens in all IPv6 interfaces, but may also listen in all IPv4
# interfaces depending on the operating system. You can specify ports with
# "host:port".
imap_listen = [::]
pop3_listen = [::]

# IP or host address where to listen in for SSL connections. Defaults
# to above non-SSL equilevants if not specified.
imaps_listen = [::]
pop3s_listen = [::]

# Disable SSL/TLS support.
#ssl_disable = no

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self-signed
# certificate, just make sure to update the domains in dovecot-openssl.cnf
ssl_cert_file = /usr/share/ssl/certs/dovecot.pem
ssl_key_file = /usr/share/ssl/private/dovecot.pem

# SSL parameter file. Master process generates this file for login processes.
# It contains Diffie Hellman and RSA parameters.
#ssl_parameters_file = /var/run/dovecot/ssl-parameters.dat

# How often to regenerate the SSL parameters file. Generation is quite CPU
# intensive operation. The value is in hours, 0 disables regeneration
# entirely.
#ssl_parameters_regenerate = 24

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that 127.*.*.* and
# IPv6 ::1 addresses are considered secure, this setting has no effect if
# you connect from those addresses.
#disable_plaintext_auth = yes

# Use this logfile instead of syslog(). /dev/stderr can be used if you want to
# use stderr for logging (ONLY /dev/stderr - otherwise it is closed).
#log_path =

# For informational messages, use this logfile instead of the default
#info_log_path =

# Prefix for each line written to log file. % codes are in strftime(3)
# format.
#log_timestamp = "%b %d %H:%M:%S "

##
## Login processes
##

# Directory where authentication process places authentication UNIX sockets
# which login needs to be able to connect to. The sockets are created when
# running as root, so you don't have to worry about permissions. Note that
# everything in this directory is deleted when Dovecot is started.
login_dir = /var/run/dovecot-login
```

```
# chroot login process to the login_dir. Only reason not to do this is if you
# wish to run the whole Dovecot without roots.
# http://wiki.dovecot.org/Rootless
#login_chroot = yes


##
## IMAP login process
##

login = imap

# Executable location.
#login_executable = /usr/libexec/dovecot/imap-login

# User to use for the login process. Create a completely new user for this,
# and don't use it anywhere else. The user must also belong to a group where
# only it has access, it's used to control access for authentication process.
# and don't use it anywhere else. The user must also belong to a group where
# only it has access, it's used to control access for authentication process.
# Note that this user is NOT used to access mails.
# http://wiki.dovecot.org/UserIds
#login_user = dovecot

# Set max. process size in megabytes. If you don't use
# login_process_per_connection you might need to grow this.
#login_process_size = 32

# Should each login be processed in it's own process (yes), or should one
# login process be allowed to process multiple connections (no)? Yes is more
# secure, espcially with SSL/TLS enabled. No is faster since there's no need
# to create processes all the time.
#login_process_per_connection = yes

# Number of login processes to create. If login_process_per_user is
# yes, this is the number of extra processes waiting for users to log in.
#login_processes_count = 3

# Maximum number of extra login processes to create. The extra process count
# usually stays at login_processes_count, but when multiple users start logging
# in at the same time more extra processes are created. To prevent fork-bombing
# we check only once in a second if new processes should be created - if all
# of them are used at the time, we double their amount until limit set by this
# setting is reached. This setting is used only if login_process_per_use is yes.
#login_max_processes_count = 128

# Maximum number of connections allowed in login state. When this limit is
# reached, the oldest connections are dropped. If login_process_per_user
# is no, this is a per-process value, so the absolute maximum number of users
# logging in actually login_processes_count * max_logging_users.
#login_max_logging_users = 256

##
## POP3 login process
##

# Settings default to same as above, so you don't have to set anything
# unless you want to override them.

login = pop3

# Exception to above rule being the executable location.
#login_executable = /usr/libexec/dovecot/pop3-login
```

```
##
## Mail processes
##

# Maximum number of running mail processes. When this limit is reached,
# new users aren't allowed to log in.
#max_mail_processes = 1024

# Show more verbose process titles (in ps). Currently shows user name and
# IP address. Useful for seeing who are actually using the IMAP processes
# (eg. shared mailboxes or if same uid is used for multiple accounts).
#verbose_proctitle = no

# Show protocol level SSL errors.
#verbose_ssl = no

# Valid UID range for users, defaults to 500 and above. This is mostly
# to make sure that users can't log in as daemons or other system users.
# Note that denying root logins is hardcoded to dovecot binary and can't
# to make sure that users can't log in as daemons or other system users.
# Note that denying root logins is hardcoded to dovecot binary and can't
# be done even if first_valid_uid is set to 0.
first_valid_uid = 500
last_valid_uid = 0

# Valid GID range for users, defaults to non-root/wheel. Users having
# non-valid GID as primary group ID aren't allowed to log in. If user
# belongs to supplementary groups with non-valid GIDs, those groups are
# not set.
first_valid_gid = 1
last_valid_gid = 0

# Grant access to these extra groups for mail processes. Typical use would be
# to give "mail" group write access to /var/mail to be able to create dotlocks.
#mail_extra_groups =

# ':' separated list of directories under which chrooting is allowed for mail
# processes (ie. /var/mail will allow chrooting to /var/mail/foo/bar too).
# This setting doesn't affect login_chroot or auth_chroot variables.
# WARNING: Never add directories here which local users can modify, that
# may lead to root exploit. Usually this should be done only if you don't
# allow shell access for users. See doc/configuration.txt for more information.
# may lead to root exploit. Usually this should be done only if you don't
# allow shell access for users. See doc/configuration.txt for more information.
#valid_chroot_dirs =

# Default chroot directory for mail processes. This can be overridden by
# giving /./ in user's home directory (eg. /home/./user chroots into /home).
#mail_chroot =

# Default MAIL environment to use when it's not set. By leaving this empty
# dovecot tries to do some automatic detection as described in
# doc/mail-storages.txt. There's a few special variables you can use:
#
#   %u - username
#   %n - user part in user@domain, same as %u if there's no domain
#   %d - domain part in user@domain, empty if user there's no domain
#   %h - home directory
#
# You can also limit a width of string by giving the number of max. characters
# after the '%' character. For example %1u gives the first character of
# username. Some examples:
```

```
#
#   default_mail_env = maildir:/var/mail/%1u/%u/Maildir
#   default_mail_env = mbox:~/mail/:INBOX=/var/mail/%u
#   default_mail_env = mbox:/var/mail/%d/%n/:INDEX=/var/indexes/%d/%n
#
#default_mail_env = mbox:~/mail:INBOX=/var/mail/%u
default_mail_env = mbox:/var/mail/%u
# Space-separated list of fields to cache for all mails. Currently these
# fields are allowed followed by a list of commands they speed up:
#
#  Envelope      - FETCH ENVELOPE and SEARCH FROM, TO, CC, BCC, SUBJECT,
#                  SENTBEFORE, SENTON, SENTSINCE, HEADER MESSAGE-ID,
#                  HEADER IN-REPLY-TO
#  Body          - FETCH BODY
#  Bodystructure - FETCH BODY, BODYSTRUCTURE
#  MessagePart   - FETCH BODY[1.2.3] (ie. body parts), RFC822.SIZE,
#                  SEARCH SMALLER, LARGER, also speeds up BODY/BODYSTRUCTURE
#                  generation. This is always set with mbox mailboxes, and
#                  also default with Maildir.
#
# Different IMAP clients work in different ways, that's why Dovecot by default
# only caches MessagePart which speeds up most operations. Whenever client
# does something where caching could be used, the field is automatically marked
# to be cached later. For example after FETCH BODY the BODY will be cached
# for all new messages. Normally you should leave this alone, unless you know
# what most of your IMAP clients are. Caching more fields than needed makes
# the index files larger and generate useless I/O.
#
# With maildir there's one extra optimization - if nothing is cached, indexing
# the maildir becomes much faster since it's not opening any of the mail files.
# This could be useful if your IMAP clients access only new mails.

#mail_cache_fields = MessagePart

# Space-separated list of fields that Dovecot should never set to be cached.
# Useful if you want to save disk space at the cost of more I/O when the fields
# needed.
#mail_never_cache_fields =

# Workarounds for various client bugs:
#   oe6-fetch-no-newmail:
#     Never send EXISTS/RECENT when replying to FETCH command. Outlook Express
#     seems to think they are FETCH replies and gives user "Message no longer
#     in server" error. Note that OE6 still breaks even with this workaround
#     if synchronization is set to "Headers Only".
#   outlook-idle:
#     Outlook and Outlook Express never abort IDLE command, so if no mail
#     arrives in half a hour, Dovecot closes the connection. This is still
#     fine, except Outlook doesn't connect back so you don't see if new mail
#     arrives.
#   outlook-pop3-no-nuls:
#     Outlook and Outlook Express hang if mails contain NUL characters.
#     This setting replaces them with 0x80 character.
#client_workarounds =

# Dovecot can notify client of new mail in selected mailbox soon after it's
# received. This setting specifies the minimum interval in seconds between
# new mail notifications to client - internally they may be checked more or
# less often. Setting this to 0 disables the checking.
# NOTE: Evolution client breaks with this option when it's trying to APPEND.
#mailbox_check_interval = 0

# Like mailbox_check_interval, but used for IDLE command.
```

```
#mailbox_idle_check_interval = 30

# Allow full filesystem access to clients. There's no access checks other than
 what the operating system does for the active UID/GID. It works with both
# maildir and mboxes, allowing you to prefix mailboxes names with eg. /path/
# or ~user/.
#mail_full_filesystem_access = no

# Maximum allowed length for custom flag name. It's only forced when trying
# to create new flags.
#mail_max_flag_length = 50

# Save mails with CR+LF instead of plain LF. This makes sending those mails
# take less CPU, especially with sendfile() syscall with Linux and FreeBSD.
# But it also creates a bit more disk I/O which may just make it slower.
#mail_save_crlf = no

# Use mmap() instead of read() to read mail files. read() seems to be a bit
# faster with my Linux/x86 and it's better with NFS, so that's the default.
#mail_read_mmaped = no

# By default LIST command returns all entries in maildir beginning with dot.
# Enabling this option makes Dovecot return only entries which are directories.
# This is done by stat()ing each entry, so it causes more disk I/O.
# (For systems setting struct dirent->d_type, this check is free and it's
# done always regardless of this setting)
#maildir_stat_dirs = no

# Copy mail to another folders using hard links. This is much faster than
# actually copying the file. This is problematic only if something modifies
# the mail in one folder but doesn't want it modified in the others. I don't
# know any MUA which would modify mail files directly. IMAP protocol also
# requires that the mails don't change, so it would be problematic in any case.
# If you care about performance, enable it.
#maildir_copy_with_hardlinks = no

# Check if mails' content has been changed by external programs. This slows
# down things as extra stat() needs to be called for each file. If changes are
# noticed, the message is treated as a new message, since IMAP protocol
# specifies that existing messages are immutable.
#maildir_check_content_changes = no

# Which locking methods to use for locking mbox. There's three available:
#  dotlock: Create <mailbox>.lock file. This is the oldest and most NFS-safe
#           solution. If you want to use /var/mail/ like directory, the users
#           will need write access to that directory.
#  fcntl  : Use this if possible. Works with NFS too if lockd is used.
#  flock  : May not exist in all systems. Doesn't work with NFS.
#
# You can use both fcntl and flock too; if you do the order they're declared
# with is important to avoid deadlocks if other MTAs/MUAs are using both fcntl
# and flock. Some operating systems don't allow using both of them
# simultaneously, eg. BSDs. If dotlock is used, it's always created first.
mbox_locks = fcntl

# Should we create dotlock file even when we want only a read-lock? Setting
# this to yes hurts the performance when the mailbox is accessed simultaneously
# by multiple processes, but it's needed for reliable reading if no other
# locking methods are available.
#mbox_read_dotlock = no

# Maximum time in seconds to wait for lock (all of them) before aborting.
#mbox_lock_timeout = 300
```

```
# If dotlock exists but the mailbox isn't modified in any way, override the
# lock file after this many seconds.
#mbox_dotlock_change_timeout = 30

# umask to use for mail files and directories
#umask = 0077

# Drop all privileges before exec()ing the mail process. This is mostly
# meant for debugging, otherwise you don't get core dumps. Note that setting
# this to yes means that log file is opened as the logged in user, which
# might not work. It could also be a small security risk if you use single UID
# for multiple users, as the users could ptrace() each others processes then.
#mail_drop_priv_before_exec = no

##
## IMAP process
##

# Executable location
#imap_executable = /usr/libexec/dovecot/imap

# Set max. process size in megabytes. Most of the memory goes to mmap()ing
# files, so it shouldn't harm much even if this limit is set pretty high.
#imap_process_size = 256

# Support for dynamically loadable modules.
#imap_use_modules = no
#imap_modules = /usr/lib/dovecot/imap

##
## POP3 process
##

# Executable location
#pop3_executable = /usr/libexec/dovecot/pop3

# Set max. process size in megabytes. Most of the memory goes to mmap()ing
# files, so it shouldn't harm much even if this limit is set pretty high.
#pop3_process_size = 256

# Support for dynamically loadable modules.
#pop3_use_modules = no
#pop3_modules = /usr/lib/dovecot/pop3
#pop3_modules = /usr/lib/dovecot/pop3

##
## Authentication processes
##

# An Authentication process is a child process used by Dovecot that
# handles the authentication steps. The steps cover an authentication
# mechanism (auth_mechanisms, how the client authenticates in the IMAP or
# POP3 protocol), which password database should be queried (auth_passdb),
# and which user database should be queried (auth_userdb, to obtain
# UID, GID, and location of the user's mailbox/home directory).
#
# You can have multiple processes, though a typical configuration will
# have only one. Each time "auth = xx" is seen, a new process
# definition is started. The point of multiple processes is to be able
# to set stricter permissions. (See auth_user below.)
#
# Just remember that only one Authentication process is asked for the
```

```
# password, so you can't have different passwords accessible through
# different process definitions (unless they have different
# auth_mechanisms, and you're ok with having different password for
# each mechanisms).
# auth_mechanisms, and you're ok with having different password for
# each mechanisms).

# Authentication process name.
auth = default

# Specifies how the client authenticates in the IMAP protocol.
# Space separated list of permitted authentication mechanisms:
#   anonymous plain digest-md5 cram-md5
#
# anonymous - No authentication required.
# plain - The password is sent as plain text. All IMAP/POP3 clients
#  support this, and the password can be encrypted by Dovecot to match
#  any of the encryption schemes used in password databases.
# digest-md5 and cram-md5 - both encrypt the password so it is more
#  secure in transit, but are not well supported by clients, and
#  require that the password database use a matching encryption
#  scheme (or be in plaintext).
#
# See auth.txt for more details.
#
# If you are using SSL there is less benefit to digest-md5 and
# cram-md5 as the communication is already encrypted.
auth_mechanisms = plain

# Space separated list of realms for SASL authentication mechanisms that need
# them. You can leave it empty if you don't want to support multiple realms.
# Many clients simply use the first one listed here, so keep the default realm
# first.
#auth_realms =

# Default realm/domain to use if none was specified. This is used for both
# SASL realms and appending @domain to username in plaintext logins.
#auth_default_realm =

# User database specifies where mails are located and what user/group IDs
# own them. For single-UID configuration use "static".
# http://wiki.dovecot.org/Authentication
# http://wiki.dovecot.org/VirtualUsers
#   passwd: /etc/passwd or similiar, using getpwnam()
#   passwd-file <path>: passwd-like file with specified location
#   static uid=<uid> gid=<gid> home=<dir template>: static settings
#   vpopmail: vpopmail library
#   ldap <config path>: LDAP, see doc/dovecot-ldap.conf
#   pgsql <config path>: a PostgreSQL database, see doc/dovecot-pgsql.conf
auth_userdb = passwd

# Password database specifies only the passwords for users.
# http://wiki.dovecot.org/Authentication
#   passwd: /etc/passwd or similiar, using getpwnam()
#   shadow: /etc/shadow or similiar, using getspnam()
#   pam [<service> | *]: PAM authentication
#   passwd-file <path>: passwd-like file with specified location
#   vpopmail: vpopmail authentication
#   ldap <config path>: LDAP, see doc/dovecot-ldap.conf
#   pgsql <config path>: a PostgreSQL database, see doc/dovecot-pgsql.conf
auth_passdb = pam

#auth_userdb = passwd-file /etc/passwd
```

```
#auth_passdb = passwd-file /etc/passwd
#auth_executable = /usr/libexec/dovecot/dovecot-auth

# Set max. process size in megabytes.
#auth_process_size = 256
# User to use for the process. This user needs access to only user and
# password databases, nothing else. Only shadow and pam authentication
# requires roots, so use something else if possible. Note that passwd
# authentication with BSDs internally accesses shadow files, which also
# requires roots. Note that this user is NOT used to access mails.
# That user is specified by auth_userdb above.
auth_user = root

# Directory where to chroot the process. Most authentication backends don't
# work if this is set, and there's no point chrooting if auth_user is root.
#auth_chroot =

# Number of authentication processes to create
#auth_count = 1

# List of allowed characters in username. If the user-given username contains
# a character not listed in here, the login automatically fails. This is just
# an extra check to make sure user can't exploit any potential quote escaping
# vulnerabilities with SQL/LDAP databases. If you want to allow all characters,
# set this value to empty.
@
# vulnerabilities with SQL/LDAP databases. If you want to allow all characters,
# set this value to empty.
#auth_username_chars =
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890.-_@

# Username to use for users logging in with ANONYMOUS SASL mechanism
#auth_anonymous_username = anonymous

# More verbose logging. Useful for figuring out why authentication isn't
# working.
#auth_verbose = no

# Even more verbose logging for debugging purposes. Shows for example SQL
# queries.
#auth_debug = no

# digest-md5 authentication process. It requires special MD5 passwords which
# /etc/shadow and PAM doesn't support, so we never need roots to handle it.
# Note that the passwd-file is opened before chrooting and dropping root
# privileges, so it may be 0600-root owned file.

#auth = digest_md5
#auth_mechanisms = digest-md5
#auth = digest_md5
#auth_mechanisms = digest-md5
#auth_realms =
#auth_userdb = passwd-file /etc/passwd.imap
#auth_passdb = passwd-file /etc/passwd.imap
#auth_user = imapauth
#auth_chroot =

# if you plan to use only passwd-file, you don't need the two auth processes,
# simply set "auth_methods = plain digest-md5"
```

SpamAssassin

File di configurazione di SpamAssassin `/etc/mail/spamassassin/local.cf`

```
# These values can be overridden by editing ~/.spamassassin/user_prefs.cf
# (see spamassassin(1) for details)

# These should be safe assumptions and allow for simple visual sifting
# without risking lost emails.

# Imposto la soglia superata la quale si considera il messaggio come spam
required_hits 5

# Riscrivo l'oggetto con la stringa SPAM
rewrite_header Subject [SPAM]

# Imposto le lingue da cui l'utente si aspetta le email
ok_languages it en
ok_local it

# Permetto agli utenti di definire proprie regole
allow_user_rules 1

# Aumento il punteggio assegnato per la parola TONER
score TONER     2.5

# Aumenta il punteggio assegnato per lingue non desiderate
score UNDESIRED_LANGUAGE BODY   2

#Aumento il punteggio dei messaggi contenenti Undisclosed-recipients
score UNDISC_RECIPS     1.5

# Aumento il punteggio dei messaggi formattati in HTML
score HTML_MESSAGE      1.5

# Inserisco una regola per messaggi il cui subject inizia con [OT]
header ANTI_TROLL Subject =~ /^\[OT\]/
describe ANTI_TROLL Subject: found OT tag
score ANTI_TROLL        3.5

# Inserisco in whitelist tutti gli indirizzi appartenenti al dominio labreti4.it
whitelist_from  *@labreti4.it

# Inserisco in blacklist l'indirizzo pluto@paperino.it
blacklist_from  pluto@paperino.it

# Aumento il punteggio per la parola VIAGRA
body VIAGRA     /viagra/i
score VIAGRA    7

# Imposto il cambiamento del messaggio se viene targato come spam
report safe     1

# Disabilito il salto del controllo sulle RBL
skip_rbl_checks 0

# Imposto a 10 il limite di tempo massimo per effettuare il controllo sulle RBL
rbl_timeout 10

# Abilito il controllo con I filtri bayesiani
use_bayes       1
use_bayes_rules 1
```

```
# Disabilito l'auto learning
bayes_auto_learn        0

# Imposto la variabile per il database dei filtri bayesiani
bayes_path /var/spool/spamassassin/bayes
```

18. RIFERIMENTI BIBLIOGRAFICI

Libri e dispense:

"Internet e Reti di Calcolatori 2/ed" James F. Kurose, Keith W. Ross - McGraw-Hill, 2003

"Bloking Spam & Spyware for Dummies" Peter Gregory, Michael Simon - Wiley Publishing, 2005

Materiale on-line:

http://taurus.polito.it/~lioy/01gsd/emailSec_2x.pdf "Sicurezza della posta elettronica – Politecnico di Torino, Antonio Lioy"

http://openskills.info "Sito di informazione sul mondo Linux"

http://en.wikipedia.org/wiki/Bayesian_filtering "Enciclopedia on-line"

http://a2.pluto.it/ "Appunti di informatica libera"

http://www.collinelli.net/antispam/ "Pagina Antispam in italiano"

http://www.studiocappello.it/Articoli_web_ed_email_marketin/articoli_email_marketing.html "Articoli e-mail marketing"

http://www.faqs.org/rfcs/rfc2821.html "RFC 2821 Protocollo SMTP"

http://www.faqs.org/rfcs/rfc1939.html "RFC 1939 Protocollo POP"

http://www.faqs.org/rfcs/rfc2060.html "RFC 2060 Protocollo IMAP"

http://www.spamterminator.it/spam.asp "Cos'è lo spam?"

http://pro.html.it/articoli/id_461/idcat_31/pag_1/pag.html "Un indirizzo e-mail anti-spam"

http://www.shinynews.it/economy/0804-spamming.shtml "Categorie di spam"

http://www.f-secure.com/2005/1/ "Rapporto F-secure"

http://punto-informatico.it/p.asp?i=41578&p=1&r=PI "Dossier/ Spaghetti spam, italiani e piaga globale"

http://www.pcself.com/sicurezza/combattere_lo_spam.html "Decalogo antispam"

http://www.paulgraham.com/spam.html "Sito di Paul Graham sui filtri bayesiani"

http://www.kensan.it/articoli/Filtri_Bayesiani.html "Filtri bayesiani"

http://www.apogeonline.com/webzine/2003/05/28/01/200305280101 "Filtri bayesiani, arma letale antispam"

http://www.weekit.it/art006004038433.jsp "Il calcolo probabilistico contro gli e-scocciatori"

http://www.wowarea.com/italiano/aiuto/spamit.htm "Spamming: come combattere gli spammer e la posta indesiderata"

http://www.tuxjournal.net/news/00263.html "Nuovo metodo di Email-Authentication"

http://sicurezza.html.it/articoli.asp?idcatarticoli=9&Idarticoli=41&npagina=1 "Tre programmi gratuiti antispam"

http://linux.html.it/articoli2/spamassassin_11.htm “SpamAssasin”
http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-it-4/ch-email.html “Guida MTA-MUA-MDA”
http://erlug.linux.it/linuxday/2002/contrib/rabbi-ld2002.html “MTA”

Riviste:

Articoli su riviste dedicate al mondo Linux:

Linux & C. anno 6 numero 40-41-42, anno 2 numero 5

- ✓ “Qmail il server di posta sicuro” di Luca Gibelli
- ✓ “Postfix il mailserver facile, veloce e sicuro” di Matteo Garofano
- ✓ “Posta elettronica: confronto tra client e-mail per Linux” di Andrea Scrimieri
- ✓ “ClamAV e SpamAssassin: integrazione con un MTA” di Luca Gibelli

Red Hat Magazine anno 1 numero 6, anno 2 numero 3

- ✓ “Stop allo spam – Configurazione e gestione di SpamAssassin” di Ralf Spenneberg
- ✓ “Installazione di un web/mail server” di Christian Ruettimann

Linux pratico anno 3 numero 10

- ✓ “No allo spam: eliminarlo con SpamAssassin” di Alessandro Dalvit

Io Programmo anno 9 numero 4

- ✓ “Filtri bayesiani contro la posta spazzatura” di Massimiliano Bigatti