Il server POP3, un servizio storico che ci permette di prelevare i messaggi dalla nostra "cassetta della posta elettronica". Tutti lo usano ma pochi raccontano come funziona e quali caratteristiche ha

Le cassette della posta e i servizi di prelievo

Matteo Garofano

m.garofano@oltrelinux.com
Amministratore di sistema presso un Internet Service Provider, si occupa di networking e sicurezza. Utilizza volentieri Slackware, NetBSD e Python.

Il servizio di posta elettronica è noto essere uno dei più utilizzati della rete e permette lo scambio di informazioni in tempi estremamente rapidi. Il suo funzionamento però, a differenza dei recenti sistemi di instant messaging, assomiglia pienamente a quello della posta tradizionale. L'infrastruttura che ne presiede il buon funzionamento si articola di vari componenti e protocolli: in generale un mittente genera un messaggio, contatta un servizio di trasferimento dei messaggi (MTA, Mail Transfer Agent) affidando ad esso quanto generato; il sistema MTA si occupa di recapitare il messaggio ad un altro server di posta e, quest'ultimo, mantiene memorizzato il messaggio così come la posta tradizionale rimane inserita nella "cassetta della posta" del destinatario.

In questo articolo e nei eventuali seguenti ci occuperemo dell'ultimo tratto di percorso che compie il messaggio, cioè dalla cassetta della posta fino alla scrivania dell'utente.

Si prenderanno in esame i differenti metodi che possono essere utilizzati per fornire il servizio di prelievo: il più noto tra questi utilizza il protocollo POP3 mentre il suo più moderno successore, non diffusamente utilizzato, si basa sul protocollo IMAP.

Prima di passare alla configurazione dei servizi faremo un breve salto nel passato per inquadrare come si sia arrivati allo sviluppo di tali protocolli.

La storia

L'architettura del servizio POP3 deriva dalla storia dello sviluppo di Internet e del funzionamento e l'evoluzione delle reti e, per questo motivo, alcune sue funzionalità oggi possono sembrare superate - e in alcuni casi lo sono - perché sviluppate in un periodo in cui le tecnologie e le disponibilità di risorse erano sostanzialmente diverse.

Senza narrare l'intera storia dello sviluppo della grande rete e dei servizi di posta elettronica si possono ricordare gli elementi che hanno influito e condizionato le caratteristiche del servizio POP3. Durante le primissime fasi dello sviluppo di Internet, agli inizi degli anni '60, lo scambio di messaggistica tra utenti avveniva attraverso mainframe in time-sharing. Utilizzando queste macchine, a cui gli utenti si connettevano solo temporaneamente attraverso linee dial-up, era possibile salvare semplici file di testo in una cartella condivisa che permetteva così lo scambio di messaggi.

Uno dei primi programmi e protocolli usati per lo scambio di messaggistica fu lo **UUCP**: si tratta, come suggerisce lo stesso acronimo di "**U**nix to **U**nix **C**o**P**y", di un sistema che consente la copia di file tra sistemi Unix. La connessione tra computer avveniva attraverso linea seriale o attraverso l'uso di modem e linee telefoniche.

Il programma UUCP, ancora attualmente disponibile e utilizzabile anche su Linux, consente di specificare un percorso del file da copiare e una destinazione che può rappresentare gli host che devono essere attraversati affinché il messaggio raggiunga la destinazione secondo un tragitto hop-by-hop, a passi. Tale percorso viene chiamato anche *bang path* poiché gli host sono separati da punti esclamativi o *bang* (ad esempio: `bighost!hostvax!thebox!user1`). Dall'host `bighost` i messaggi passavano all'host `hostvax`, da questi alla macchina `thebox` dove si trovava l'utente `user1` che poteva leggere i suoi messaggi.

Solo alla fine degli anni '60 vengono connessi i primi computer in rete e nei primi anni '70

vedono la luce i primi protocolli di comunicazione host-to-host. La crescita di Internet è inizialmente affidata ad istituzioni che possono contare su infrastrutture all'avanguardia e su linee di connessione permanenti, anche se cominciano ad aumentare gli host che riescono ad essere raggiungibili solo per un breve periodo di tempo (accesso temporaneo).

In alcuni host di piccole dimensioni è spesso impossibile mantenere attivo un sistema di messaggistica (**MTS**, Message Transport System): le ragioni sono diverse, e vanno dalla mancanza di risorse computazionali alla mancanza di storage (ricordiamo che sebbene si parli soltanto di decine di anni si tratta di alcune "ere" informatiche indietro), ma è soprattutto un link attivo in modo permanente ad impedire la possibilità di ricevere la posta in tempo reale. Per questi motivi, già nelle prime fasi di sviluppo del sistema di scambio della posta elettronica, nasce l'esigenza di poter disporre di un sistema che consentisse di prelevare la posta da host remoti.

Questo è uno dei punti cardine che hanno condizionato le decisioni sulla scelta delle caratteristiche che dovevano avere i sistemi di scambio di messaggi.

A tal scopo venne pensata una soluzione che consentisse di mantenere la posta per un tempo indeterminato e di permetterne il prelievo da parte del destinatario finale; il servizio prese il nome di Post Office Protocol che abbreviato diventerà **POP**. Dalle sue varie versioni, basate sulle prime sperimentazioni risalenti alla metà degli anni '80, il protocollo POP è arrivato alla versione 3, da cui il famoso parametro POP3 presente nella quasi totalità dei client di posta odierni.

Architettura

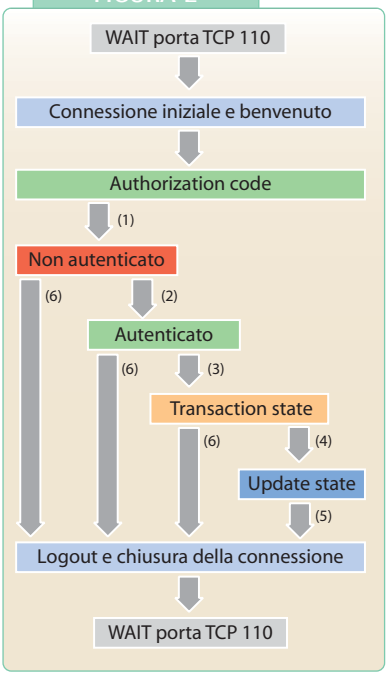
L'architettura sottesa al prelievo dei messaggi attraverso il protocollo POP3 è quella tipica client-server.

Una sessione POP3 si sviluppa attraverso una serie di stati diversi: una volta che la connessione TCP è stata aperta e il server POP3 ha inviato al client il suo messaggio di benvenuto, si entra nello stato di **autenticazione** dove il client deve farsi identificare dal server POP3; essa, in caso di successo, conduce allo stato di **transizione**, nel quale il server alloca le risorse per poter accedere alla mailbox e il client può richiedere lo svolgimento di azioni da parte del server.

Quando il client invia il comando `QUIT` si passa allo stato di **update**: le risorse riservate per le operazioni precedenti vengono rilasciate, vengono eseguiti i comandi di cancellazione precedentemente impartiti e la connessione TCP viene chiusa.

Lo stato di autenticazione può essere superato con due procedure diverse: di norma utilizzano i due comandi `user` e `pass` aventi come argomenti rispettivamente username e password

FIGURA 2



Stati di una sessione POP3

(1) connessione avvenuta con successo (OK messaggio di benvenuto)

(2) LOGIN avvenuta o comando di autenticazione

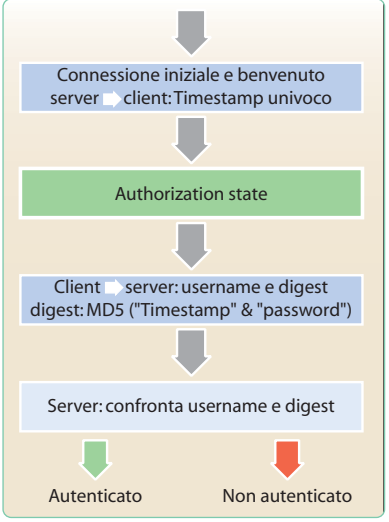
(3) operazione di SELECT avvenuta o comando di EXAMINE

(4) comando di chiusura QUIT del client

(5) aggiornamento della mailbox

(6) comando di LOGOUT, shutdown del server o chiusura della connessione per interruzione o timeout

FIGURA 1



Il flusso di scambio dati usando l'autenticazione APOP

diventando di fatto uno scambio di credenziali tra client e server. Questa procedura comporta il passaggio delle credenziali *in chiaro* sulla rete e l'esposizione alla possibile cattura attraverso strumenti di sniffing. Una procedura alternativa, più sicura, prevede l'utilizzo del comando `APOP`: se il server supporta il metodo di autenticazione APOP, nel messaggio di benvenuto viene incluso un timestamp, ovvero un numero univoco legato al tempo che sarà diverso per ciascun messaggio di benvenuto che viene emesso. Il client memorizza il valore restituito dal

server ed invia il comando APOP utilizzando due parametri per l'identificazione, una username e un digest: lo username è quello classico presente anche nel POP3, ed è relativo ad un certo utente. Il parametro digest viene invece costruito applicando la funzione MD5 alla stringa timestamp inviata dal server seguita da un segreto conosciuto da client e server (tipicamente la password): il server confronta il digest ricevuto dal client con quello da lui calcolato e, nel caso di corrispondenza, passa allo stato di transizione.

Se anche vi fosse stato sniffing durante la sessione, l'attaccante avrebbe solo la conoscenza dell'username, e dovrebbe effettuare tentativi a forza bruta per quanto riguarda il digest: quello inviato, infatti, è legato alla sessione in cui è stato generato (c'è un timestamp) e quindi non più utilizzabile in futuro.

I comandi che si possono impartire al server POP3 durante lo stato di transizione sono parole dove non vengono distinte maiuscole e minuscole ("case-insensitive" quindi), eventualmente seguiti con un argomento. I comandi `USER`, `PASS`, `APOP` sono considerati opzionali.

All'invio dei comandi segue un responso che può essere di due tipi: `+OK` oppure `-ERR`.

Comandi opzionali, estensioni e capabilities

Con il passare degli anni, le esigenze degli utenti sono cresciute pertanto si è cercato di ovviare ai limiti imposti dal protocollo POP che abbiamo descritto. Per questa ragione, durante la lunga storia di questo protocollo, sono stati aggiunti comandi opzionali con i quali si è cercato di superare tali limitazioni aggiungendo nuove funzionalità che possono essere definite, a tutti gli effetti, estensioni del protocollo POP3.

Per evitare confusione è stato introdotto il concetto di *capabilities* intendendo un metodo per rendere noti quali siano i comandi opzionali e le estensioni implementate e rese disponibili dal server. Con il comando `CAPA`, utilizzabile negli stati di autenticazione e di transizione, si accede a tali informazioni. Queste funzionalità sono descritte dal documento RFC 2449.

Le estensioni che possono essere rese disponibili sono numerose e i server POP3 moderni le forniscono così come la maggior parte di client ne sfrutta le funzionalità.

Il protocollo POP3 però non nasce con l'idea di manipolare in modo complesso la posta, ma solo di scaricarla sul computer locale e, da qua, gestirla al meglio. Nonostante questo sia lo scopo dichiarato, è possibile lasciare una parte o tutta la posta sul server anche dopo il download della stessa: affinchè vengano riconosciute le mail già scaricate da quelle ancora da leggere, è stato implementato un identificativo univoco per ogni messaggio, chiamato **UID** (**U**nique **ID**entification), generato dal server per ogni mail e tipicamente memorizzato dal client di posta (il cosiddetto **MUA**, **M**ail **U**ser **A**gent): tramite questo hash persistente (che può essere lungo fino a 70 caratteri), vengono identificati dal client i messaggi già letti da quelli ancora da leggere.

I diversi server POP3 implementano in modi distinti la funzione di generazione dell'identificativo UID, e questo è uno degli

TABELLA 1

COMANDO *argomento*	Stato	Cosa fa?
USER *username*	Autenticazione	Permette di passare l'identificativo dell'utente
PASS *password*	Autenticazione	Permette di passare le credenziali (segreto) dell'utente
APOP *username digest*	Autenticazione, opzionale	Permette di autenticare attraverso una coppia username/digest (MD5), vedi testo
STAT	Transizione	Da informazioni sullo stato della mailbox per esempio `"+OK 2 5120"` indica la presenza di 2 messaggi e della dimensione della mailbox di 5120 Byte
LIST	Transizione	Consente di elencare il numero di messaggi ad es. `"+OK 1 visible messages (567 octets) 1 567"` dove `1` è il numero di messaggi e `567` il numero di byte
RETR *n*	Transizione	Visualizza il messaggio numerato *n*
DELE *n*	Transizione	Cancella il messaggio numerato *n*
NOOP	Transizione	Il server non fa null'altro che rispondere con una risposta positiva
RSET	Transizione	Se qualche messaggio era stato segnato come "da cancellare" tale impostazione viene annullata
QUIT	Transizione	Consente di chiudere la connessione

TABELLA 2

COMANDO *argomento*	Stato	Cosa fa?
CAPA		Visualizza le Capabilities ovvero le funzioni rese disponibili
RESP-CODES		Le risposte del server vengono fornite con "codici estesi" anzichè i soli "successo" o "fallimento"
LOGIN-DELAY		Mostra l'intervallo minimo di tempo con cui è possibile eseguire 2 autenticazioni
PIPELINING		Mostra la capacità del server di accettare più comandi senza attendere la risposta
EXPIRE		Indica il tempo minimo con cui verranno conservati i messaggi. **0** indica che implicitamente al passaggio allo stato di update considererà da cancellare i messaggi che sono stati prelevati con il comando `RETR`
IMPLEMENTATION		

TABELLA 3

COMANDO *argomento*	Stato	Cosa fa?
TOP *msg n*	Transizione, opzionale	Consente di visualizzare le *n* linee del messaggio numerato *msg*
UIDL *[n]*	Transizione, opzionale	Visualizza l'identificativo univoco del messaggio
SASL	Autenticazione	Ovvero: **S**imple **A**uthentication and **S**ecurity **L**ayer, fornisce un sistema di autenticazione sicuro

TABELLA 4

Nome	Licenza	Mbox	Mdir	Autenticazioni	TSL/SSL	Sicurezza	Documentazione	Features peculiari
Qpopper	Specifica	Si	No/patch	Password/shadow, APOP, Kerberos 4 e 5, PAM	Si	Media	Buona	Storico server, molto maturo
popa3d	Stile BSD	Si	No/patch	Password/shadow, PAM	No	Alta	Scarsa	Piccolo impiego di memoria e sicuro
Courier-pop3d	GNU GPL	No	Si	Password/shadow, PAM, MySQL PostgreSQL	Si	Media	Scarsa	Si integra con la suite courier-mta
Teapop	Stile BSD, con copyright, no "pubblico dominio"	Si	Si	MySQL, PostgreSQL, apache htpasswd	No	Media	Sufficiente	Supporto ai domini virtuali
Dovecot-pop3d	GNU LGPL	Si	Si	Password/shadow, PAM, MySQL PostgreSQL, LDAP, APOP	Si	Alta	Buona	Server scalabile e sicuro con supporto a numerosi sistemi di autenticazione
qmail-pop3d	Specifica	No	Si	Server scalabile e sicuro con supporto a numerosi sistemi di autenticazionepassword/shadow, altro/patch, APOP	No/patch	Alta	Scarsa	Si integra con la suite qmail
tpop3d	GNU GPL	Si	Si	Password/shadow, PAM, MySQL, PostgreSQL, LDAP, flat-file, modulo custom, pipe a programmi	Si	??	Scarsa	Sistema di autenticazione modulare molto flessibile

aspetti da ricordare quando si migra un sistema da un server POP3 ad uno diverso: se lo UID cambia, perché il nuovo server non gestisce il vecchio formato e quindi ne assegna uno diverso ad ogni mail, al prossimo check della posta del client tutti i messaggi non saranno riconosciuti, quindi identificati non letti e scaricati nuovamente (cosa che accade spesso quando è il client ad andare in crash e si perde il database degli UID).

Mani sulla tastiera

La comunicazione client/server che sottente la comunicazione POP3 è totalmente in chiaro quindi, per capire un po' meglio come funziona il protocollo, potremo collegarci al server utilizzando un client specialissimo... telnet!
Provate con il vostro provider sempre che questo consenta l'accesso POP3 per la lettura della posta. Da una shell aprite la connessione verso un vostro server POP3 con il comando:

```
$ telnet pop.serverPOP3.it 110
```

a questo punto se non vi ritornano messaggi di errore siete connessi al server, comunicate le vostre credenziali con *(senza il >)*:

```
> user vostrausername
> pass vostrapassword
```

Se l'autenticazione è andata a buon fine, tramite il comando `STAT` potrete conoscere quanti messaggi contiene la mailbox

```
> stat
+OK 62 50166
```

ci indica, ad esempio, la presenza di 62 messaggi per un totale di dati di 50166 Byte, potrete ora visualizzare la lista dei messaggi con il comando:

```
> list
```

ed accedere ai messaggi stessi con il comando

```
> retr n
```

dove *n* è il numero del messaggio visualizzato nella lista. Potrete anche visualizzare le capabilities del server:

```
> capa
+OK Here's what I can do:
STLS
TOP
USER
LOGIN-DELAY 10
PIPELINING
UIDL
IMPLEMENTATION Courier Mail Server
```

E' quindi possibile utilizzare le capabilities appena scoperte: ad esempio, con `UIDL` sarà possibile conoscere l'UID di un particolare messaggio:

```
> uidl 1
+OK 1 UID31995-1123168896
```

Con il comando

```
> dele n
```

cancelliamo il messaggio che nella lista era indicato con il numero *n*: il messaggio non viene immediatamente cancellato e, per la rimozione vera e propria, bisognerà aspettare l'uscita della sessione, quando si passerà allo stato UPDATE. Possiamo ancora cambiare idea e mantenere il messaggio attraverso l'operazione

```
> rset
```

mentre, se si vuole chiudere la sessione di editing, con

```
> quit
```

potrete passare allo stato UPDATE e terminare. Fate bene attenzione perché la password non sarà oscurata durante la digitazione! Se ci sono spettatori durante queste prove potranno leggere chiaramente la password sullo schermo mentre, se siete da soli, prima di allontanarvi dalla postazione "pulite" la schermata.

Alcuni server POP3

I server POP3 opensource disponibili per l'installazione su un nostro sistema Linux sono un buon numero, per cui è forse necessaria una breve panoramica per scoprire le caratteristiche salienti di ognuno. Ovviamente, ogni server seguirà le specifiche RFC, anche perché è l'unico modo affinchè i client generici possano collegarvisi: molto della bontà del prodotto dipenderà dal supporto per le mailbox di vari formati, dalla semplicità di gestione, dalla possibilità di interfacciarsi con sistemi di autenticazione (e i loro back-end) più diversi.
Alcuni server sono molto maturi, offrono una provata stabilità e hanno un ottima diffusione anche se, in passato, hanno manifestato alcuni problemi di sicurezza: è questo il caso di Qpopper sviluppato da Qualcomm, creatrice del noto client di posta Eudora.
Altri, più recenti, si sono concentrati sugli aspetti della sicurezza, come per esempio il demone popa3d che adotta varie tecniche per raggiungere un buon livello di robustezza: tra queste vi è la minimizzavate le operazioni svolte come utente root, il controllo di qualsiasi informazione proveniente dall'esterno assunta come inaffidabile, i controlli contro i DoS.
Un altro demone POP3 molto famoso è Dovecot: il suo sviluppatore ha voluto di recente prendere l'esempio di D. J. Bernstein, autore del famoso MTA Qmail, offrendo 1.000 euro a chi riesca a trovare un bug di sicurezza (dalla versione 1.0beta2).
Alcuni sviluppatori hanno poi dedicato i loro sforzi implementativi verso la flessibilità cercando di dare il massimo spettro possibile di utilizzo supportando i formati mailbox e maildir e un gran numero di backend per l'autenticazione: questo è il caso di Dovecot-POP3d, Teapop, tPOP3d.
Altri demoni derivano da suite integrate di prodotti che cercano di coprire l'intero spettro delle attività di un mailserver rendendo disponibili tutti i servizi di MTA, di POP3 e IMAP. Tra questi figurano sicuramente il demone Courier-POP3d figlio di Courier-MTA sviluppato dalla Inter7 e qmail-pop3d appartenente al solido e ben noto server Qmail sviluppato.
Ovviamente la semplicità di configurazione ed amministrazione, la disponibilità di buona ed aggiornata documentazione, il supporto anche commerciale, la scalabilità in caso le utenze possano diventare migliaia e la disponibilità di veloci aggiornamenti in caso vengano scoperte falle o difetti sono sicuramente fondamentali per la valutazione finale del server da mettere in produzione. Per chiarire il quadro relativo ad alcuni dei più diffusi server POP3 attualmente disponibili e utilizzabili sul nostro sistema *nix si può prendere visione della *tabella 4*.

Scegliamo il server POP3

Dopo questa carrellata, c'è da decidere quale sarà il server installare: le valutazioni da fare sono diverse, ma soprattutto bisognerà fare attenzione alle caratteristiche dei componenti che vogliamo far interagire. Tre punti prenderemo in particolare considerazione:

- il formato delle caselle postali: mailbox/maildir;
- il sistema di autenticazione da usare: utenti di sistema (password/shadow), PAM, LDAP, MySQL, PostgreSQL, ecc.
- il supporto alle comunicazioni crittografate SSL/TLS.

Il nostro scenario, al momento, non sarà complesso: una piccola società con qualche decina di utenti, con un solo server di posta SMTP, POP3 e, più avanti, IMAP e webmail, e un solo dominio.
Nell'esempio utilizzeremo il moderno e sicuro server POP3 Dovecot dotato, come detto, di una ottima flessibilità sia relativa ai formati supportati sia nei confronti dei sistemi di autenticazione, che ci garantirà, nel caso in cui lo scenario cambiasse, una migrazione piuttosto semplice.
Il server memorizzerà i messaggi di posta nel classico formato mailbox, collocate nella cartella `/var/spool/mail/`*nomeutente*: per l'autenticazione ci affideremo alla coppia password/shadow di sistema, confidando sul fatto che tali utenti non avranno shell associate (in generale, su un server di posta, gli utenti interattivi

RIQUADRO 2

POP-before-SMTP: spedire mail anche da host untrusted

I server POP3 sono per la maggior parte dei casi strettamente legati al servizio SMTP. Quest'ultimo quasi certamente non consente il cosiddetto relay, cioè non permette a utenti non autorizzati di spedire messaggi verso l'esterno, questo per ostacolare l'opera degli odiati "spammer".
Il problema, in questi casi, è come identificare gli utenti "autorizzati": la cosa più semplice consiste nel protocollo SMTP autenticato, dove anche l'invio della posta non può prescindere da una autenticazione verso il server. E' una procedura, questa, non troppo standard, e si sa che meno username e password utilizzano gli utenti e più sono felici.
Nel caso più semplice sarà sufficiente autorizzare gli utenti provenienti da certi IP, ad esempio quelli di rete locale, quelli provenienti da una VPN, quelli provenienti da connessioni ad IP fisso. Molto spesso però è necessario autorizzare utenti esterni alla rete locale, poiché accedono al nostro server attraverso connessioni dial-up, magari da un portatile attraverso una rete non gestita da noi. In questi casi può essere necessario dare a tali utenti la possibilità di inviare messaggi, attraverso il protocollo SMTP, solo dopo che hanno effettuato un accesso al servizio POP3, ovvero che si sono autenticati per un altro servizio (POP e non SMTP), ma hanno dimostrato, in ogni caso, di aver diritto all'utilizzo della risorsa..
Per sfruttare tale logica ci sono numerose strade e molti programmi sono stati sviluppati.
Tra i più noti esiste uno script perl, `popbsmpt.pl`, che legge il log del demone pop3 e individua le linee di autenticazione riuscita: l'IP dell'utente autorizzato viene inserito nel database usato dal server SMTP dove sono elencate le macchine autorizzate al relay.
La concessione è solo temporanea e periodicamente dall'elenco vengono cancellate le entry scadute.
Molti sono i server supportati, alcuni, quelli con un design migliore, direttamente, altri tramite patch.

Pop-before-smtp: http://popbsmtp.sourceforge.net/
Drac: http://mail.cc.umanitoba.ca/drac/
Whoson: http://whoson.sourceforge.net/

dovrebbero essere ridotti al minimo). Vorremo inoltre consentire l'accesso attraverso un canale sicuro che utilizzi la crittografia per proteggere le nostre credenziali di accesso e i dati in transito.

Installare Dovecot POP3

Qualsiasi sia la distribuzione che state utilizzando, vi sarà sicuramente una pacchettizzazione già pronta per Dovecot: utilizzando Ubuntu Dapper, ad esempio, troviamo i binari sia per il POP3 che per l'IMAP:

```
$ apt-cache search --names-only dovecot
dovecot-common  - secure mail server that supports
                  mbox and maildir mailboxes
dovecot-imapd   - secure IMAP server that supports
                  mbox and maildir mailboxes
dovecot-pop3d   - secure POP3 server that supports
                  mbox and maildir mailboxes
```

L'installazione procede con il canonico:

```
$ sudo apt-get install dovecot-pop3d
```

Facciamo attenzione alla versione installata poiché nei nostri esempi ci riferiremo al ramo dovecot-1.x, i file di configurazione, infatti, differiscono rispetto alle release precedenti.
L'installazione, nella pacchettizzazione appena vista, creerà un utente `dovecot` con home in `/usr/lib/dovecot`, una directory `/etc/dovecot/` dove troveremo il file di configurazione `dovecot.conf`, e uno script di avvio `/etc/init.d/dovecot`; il server pop3 verrà avviato ai runlevel 2, 3, 4 e 5:

```
$ sysv-rc-conf --list dovecot
dovecot    0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

Se lo scenario è quello descritto, prima avviare il nostro server dovremo apportare solo piccole modifiche al file di configurazione di esempio che ci viene fornito ricco di commenti. In particolare le opzioni che ci occorrono sono:

```
protocols = pop3
ssl_disable = yes
log_path = /var/log/dovecot.log
login_user = dovecot
default_mail_env =
mbox:~/Mail/:INBOX=/var/spool/mail/%u
protocol pop3 {
  pop3_uidl_format = %08Xu%08Xv
  }
auth default {
mechanisms = plain
passdb shadow {
      }
      userdb passwd {
      }
user = root
}
```

Alcune delle direttive indicate nel file di configurazione sono facilmente comprensibili. In particolare con `login_user` indicheremo l'utente che servirà a far funzionare i processi di autenticazione e, come riposta sul wiki di Dovecot, non deve appartenere a nessun gruppo se non a se stesso, in particolare non deve appartenere al gruppo `mail`:
http://wiki.dovecot.org/UserIds
Poiché invece l'installazione inserisce proprio nel gruppo `mail` l'utente `dovecot`, è opportuno rimuoverlo:

```
$ sudo delgroup dovecot mail
Removing user `dovecot' from group `mail'...
done.
$ sudo id dovecot
uid=113(dovecot) gid=113(dovecot) groups=113(dovecot)
```

La direttiva `default_mail_env` indica il tipo di formato per le caselle di posta (in questo esempio, il formato mbox classico), insieme alla directory dove verranno memorizzate le stesse (`INBOX=/var/spool/mail/`): la chiave di ricerca sarà l'username dell'utente (`%u`), per cui l'utente `user01` e l'utente `user02` vedranno memorizzata la propria posta nei file `/var/spool/mail/user01` e `/var/spool/mail/user02`.
Con tali impostazioni si avrà un sistema piuttosto "classico" dove i file delle mailbox si trovano tutte in un'unica cartella, compatibilmente a quello che può essere un sistema MTA tipo Sendmail o Postfix nella configurazione di default: in caso venga rifiutato l'accesso, controllate i permessi della directory `/var/spool/mail` o `/var/mail`.
Le direttive del protocollo POP3 sono raccolte in un blocco che, nel caso dell'esempio, riguarda solo la costruzione del tag UIDL, di fondamentale importanza per il riconoscimento dei messaggi dei quali è già stato effettuato il download da parte dei client: per la composizione del tag viene effettuato un hash (`%..X`) lungo 8 byte (`.08.`) del nome utente(`....u`), unito con un hash analogo calcolato sul campo UIDValidity (`....v`).

RIQUADRO 4

Webografia

Potete trovare gli RFC diPOP3 e relative estensioni all'URL:
http://www.ietf.org/rfc/rfc*XXXX*.txt
Dove **XXXX** sta per uno dei seguenti numeri:
RFC 1939 per POP3; **RFC 2449** per le estensioni a POP3;
RFC 2246 per il protocollo TLS.
Home page di Dovecot:
http://www.dovecot.org/
Wiki e documentazione di Dovecot:
http://wiki.dovecot.org/

Segue il blocco relativo all'autenticazione: vengono specificate:

- il tipo di autenticazione in chiaro (`mechanism=plain`);
- l'archivio di username e password sono le classiche passwd/shadow utilizzate dal sistema;
- l'utente con il quale dovrà essere in esecuzione il server Dovecot: se si utilizza l'autenticazione tramite file shadow come nell'esempio, l'utente non può essere che root, altrimenti la lettura del file delle password verrebbe impedita. Comportamento analogo lo si ha utilizzando PAM mentre l'autenticazione tramite virtual user permetterebbe l'utilizzo di un utente non privilegiato.

Se ci saranno problemi potremo indagare con attenzione nei file di log di Dovecot (`/var/log/dovecot.log`) e in quelli del sistema e approfondire per cercare di capire quale sia il problema. Se la situazione è particolarmente critica si potrà avviare il server POP3 dopo aver modificato nella configurazione alcune opzioni di logging (`auth_verbose = yes`, `auth_debug = yes`).
Considerate che l'utilizzo di utenti di sistema con autenticazione in chiaro, specialmente nel caso questi abbiano anche una shell interattiva (cosa da vietare assolutamente, su un server di posta non dovrebbero esserci per alcun motivo) può essere estremamente pericoloso: meglio, in questi casi, far uso di crittografia per evitare che l'invio delle credenziali sia facilmente intercettabile. Ecco le aggiunte che saranno necessarie al file `dovecot.conf`:

```
#se vogliamo soltanto la modalità protetta possiamo
disabilitate il protocollo pop3
protocols = pop3 pop3s
ssl_disable = no
ssl_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key_file = /etc/ssl/private/ ssl-cert-snakeoil.key
```

Durante l'installazione di Dovecot, gli script contenuti nel package creeranno un certificato di default (`ssl-cert-snakeoil`) per dare la possibilità di provare il servizio: va da sé che è conveniente generarne uno personalizzato con i vostri dati, oppure di utilizzarne uno di una certification authority riconosciuta.
Effettuate le modifiche non ci resta che far ripartire il servizio

```
$ sudo /etc/init.d/dovecot restart
```

e sfruttare il collegamento al nostro server pop3 attraverso un canale protetto configurando opportunamente il nostro client di posta preferito, che dovrà connettersi sulla porta 995.