

+Imatges

- Figura 1
- Figura 2
- Figura 3
- Figura 4
- Figura 5
- Figura 6
- Figura 7
- Figura 8
- Figura 9
- Figura 10
- Figura 11
- Figura 12
- Figura 13
- Figura 14
- Figura 15
- Figura 16
- Figura 17
- Figura 18
- Figura 19
- Figura 20
- Figura 21
- Figura 22
- Figura 23
- Figura 24
- Figura 25
- Figura 26
- Figura 27
- Figura 28
- Figura 29
- Figura 30
- Figura 31
- Figura 32
- Figura 33

1. Instal·lació i administració del servei de correu electrònic

En molts conceptes el correu electrònic imita el funcionament del correu postal. És un sistema distribuït que permet als usuaris enviar missatges a un destinatari final. El correu electrònic ha tingut una important evolució des dels primers sistemes que únicament podien intercanviar missatges de text ASCII fins als correus electrònics amb continguts multimèdia d'avui en dia.

En el servei de correu es diferencia molt clarament entre el mecanisme de transport dels missatges i els missatges. El mecanisme de transport dels missatges és el protocol SMTP i és independent del format i el contingut del missatge. Els missatges originals eren en text pla ASCII de 7 bits, però actualment es permet tot tipus de contingut en un missatge. Això és possible gràcies als tipus MIME que descriuen i codifiquen els missatges.

El mateix disseny del sistema de correu ha evolucionat a mesura que ha evolucionat la tecnologia a Internet. En el model bàsic de transport usant SMTP, s'exigeix que el receptor disposi de connexió permanent a Internet i es connecti al servidor de correu localment per tal de consultar-lo. Quan els usuaris tenen accés a Internet per un ISP volen poder baixar tot el correu de cop i poder examinar-lo un cop tancada la connexió (per no pagar trucada telefònica). El protocol POP proporciona aquest mecanisme per baixar d'un servidor de correu tots els missatges de l'usuari.

Un cop Internet es popularitza i s'abaixen els costos de connexió, l'usuari s'acostuma a baixar el correu des de llocs diferents, però té l'inconvenient que el correu li queda repartit per diverses màquines (usant el correu POP). Cal un mecanisme que permeti accedir i gestionar el correu i les bústies directament en el servidor. El protocol IMAP proporciona aquest mecanisme.

Tot això ha canviat amb la popularització d'Internet a tots els nivells i amb tot tipus de dispositius. La major part del correu email es actualment via

web utilitzant proveïdors com els coneguts *Gmail*, *yahoo*, etc.

1.1.Descriu els diferents protocols que intervenen en l'enviament i recollida del correu electrònic.

El servei de correu electrònic és un dels primers serveis que es van utilitzar en les xarxes i un dels més populars a Internet. Ha tingut una important evolució des dels primers sistemes que podien intercanviar únicament missatges de text ASCII (7 bits) fins als portals web usats per milions d'usuaris per intercanviar correu amb continguts multimèdia.

El 1982 es van desenvolupar els estàndards que defineixen el correu electrònic, els quals es descriuen en el document RFC 821, que explica el protocol de transmissió, i el document RFC 822, que descriu el format dels missatges. Aquests dos estàndards han evolucionat i actualment s'utilitzen els documents RFC 2821 i RFC 2822. A més a més, per permetre missatges multimèdia s'ha definit l'estàndard MIME corresponent al document RFC 2231.

L'especificació original distingeix molt clarament entre el mecanisme de transport dels missatges i els missatges. El mecanisme de transport dels missatges és el protocol **SMTP (simple mail transport protocol o protocol simple de transport de correu)** i és independent del format i el contingut del missatge. El missatge es compon del **sobre** (o *envelop*) i el **contingut**, que al mateix temps el formen les capçaleres i el cos del missatge.

!!

Per obtenir més informació sobre l'especificació del protocol SMTP en els RFC 821, 822, 2821 i 2822, aneu a la secció "Adreces d'interès" del web del crèdit.

El correu electrònic és un sistema distribuït que permet als usuaris enviar missatges a un destinatari final. Per aconseguir-ho hi intervenen diversos agents que es descriuen a continuació:

!!

Podeu esbrinar més coses de l'estàndard MIME consultant el subapartat "Els tipus MIME" d'aquest mateix nucli d'activitat.

- **MUA (mail user agent o agent de correu d'usuari).** L'usuari utilitza un MUA per redactar, rebre i manipular correus electrònics. Un MUA és un programari que permet aquestes capacitats que poden ser aplicacions en línia d'ordres (com per exemple l'ordre Unix mail), aplicacions de text (com, per exemple, mutt, pine, etc.), interfícies gràfiques (com thunderbird) i portals de correu web (com Gmail, Yahoo, etc.). L'usuari interactua usualment amb un MUA i és aquest el que lliurarà el missatge al sistema de transport (SMTP) per fer-lo arribar al destinatari en el cas d'enviar correu, o bé el MUA obtindrà el missatge d'una bústia de correu (on l'ha dipositat l'SMTP) i el mostrarà a l'usuari en cas de recepció de correu.
- **MTA (mail transport agent o agent de transport de correu).** L'agent de transport de correu és l'encarregat de transportar els missatges al destinatari indicat. Aquesta tasca correspon al protocol **SMTP**. L'MTA rep el missatge d'un MUA i s'encarrega del seu transport fins al destinatari final. Generalment realitzen la funció de client/servidor o emissor/receptor al mateix temps. La funció que es realitza en cada cas es descriu a continuació:
- **MTA client SMTP (emissor).** S'anomena *client de correu o emissor* (segons l'arquitectura client-servidor) el servidor SMTP (fixeu-vos en l'ambigüitat) que envia el correu cap al destinatari. És qui envia el correu utilitzant el protocol SMTP. Estableix les connexions amb els servidors/receptors SMTP.
- **MTA servidor SMTP (receptor).** S'anomena *servidor de correu o receptor* el programari de servidor SMTP que rep els missatges de correu entrant i els lliura a la bústia del destinatari si es tracta d'un lliurament local, o els reenvia a un altre servidor SMTP si va destinat a un sistema remot. Fixeu-vos, per tant, que el fet que un receptor MTA rebí un correu no significa que el missatge hagi arribat al destinatari final.

Ambigüitat dels agents del sistema de correu

Els agents que intervenen en el sistema de correu electrònic tot sovint fan més d'un paper, fet que provoca ambigüitat en la definició de cada un.

Servidor SMTP

Sovint el programari de servidor SMTP (com, per exemple, *sendmail*) fa tant la funció de client (emissor de missatges) com la de servidor (receptor de missatges).

- **MDA (mail delivery agent o agent de lliurament de correu).** Un element extra en l'estructura de correu són els MDA. Són els encarregats de fer el lliurament final del missatge en la bústia del destinatari. En el procés poden realitzar diverses accions segons un conjunt de regles "*forward*" definibles. Un exemple d'MDA és el programa procmail, que permet filtrar els missatges entrants posant-los en una bústia o una altra, esborrant-los, marcant-los com a correu brossa (*spam*), fent-ne còpies, reenviant-los a altres bústies i a altres destinataris, etc. Usualment, en sistemes de correu que no disposen d'MDA, és el mateix MTA qui diposita el missatge en la bústia del destinatari final.

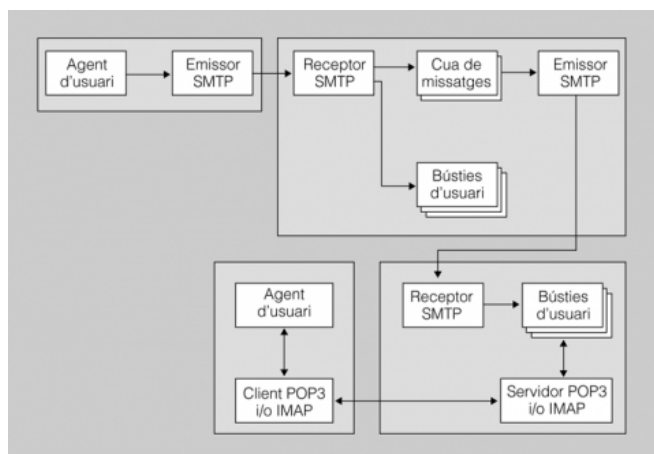
- **Adreça de correu.** Els usuaris que volen utilitzar el sistema de correu electrònic han de disposar d'una bústia de correu en un servidor de correu. Els missatges s'adreces per la xarxa utilitzant la ja coneguda nomenclatura **usuari@domini-servidor-correu**, que es llegeix *compte de correu de l'usuari tal en el domini qual*. Així, per exemple, a un usuari amb un compte de correu de nom *pere* en el domini *ioc.cat* se li poden adreçar missatges a l'adreça *pere@ioc.cat*.

- **Bústies de correu o mail box.** Els usuaris tenen bústies de correu en un servidor de correu. Quan el servidor de correu MTA rep un missatge destinat a un usuari amb compte de correu en el propi servidor, el diposita en la bústia de correu corresponent (si no hi ha un MDA pel mig). Fixeu-vos que dipositar el missatge en la bústia de l'usuari no garanteix que l'usuari el llegeixi, cal un altre pas que és la recuperació del missatge de l'usuari de la seva bústia de correu. Aquest pas es realitza des d'un MUA i sovint empra protocols com **POP** o **IMAP**, fora de l'abast de les explicacions del correu SMTP.

- **Llistes de correu i àlies.** Els àlies i les llistes de correu es tradueixen a adreces de comptes de correu. Si la llista de correu es gestiona localment el MUA local l'expandirà en el conjunt d'adreces d'usuari corresponents i enviarà el correu electrònic a cada un. Si la llista d'usuaris és remota, s'envia el correu electrònic al sistema remot i serà l'MTA remot el que l'expandirà i enviarà un correu electrònic a cada membre de la llista. Fixeu-vos que si la llista conté usuaris d'altres dominis de correu (on sigui del món) farà arribar una còpia a aquests usuaris.

El model funcional del protocol SMTP que mostra els elements que intervenen en una comunicació d'aquest tipus es pot veure en la Figura 01 "Model funcional del protocol SMTP".

Figura 01. Model funcional del protocol SMTP



Per tant, vist en conjunt, un MUA (thunderbird per exemple) serveix a l'usuari per crear un correu electrònic, i el MUA lliura el missatge a l'MTA del sistema (per exemple, sendmail) perquè el faci arribar al destinatari final. Usant el protocol SMTP el MTA s'encarrega de fer el lliurament en el sistema final (pot ser amb una connexió directa o encaminant-se a través de diversos MTA) i el missatge es diposita en la bústia de correu del receptor. En aquest procés de deixar el missatge a la bústia del receptor pot haver-hi un MDA que 'post processa' el missatge o ho pot fer directament el MTA. El receptor, quan ho considera oportú, retroba els missatges de la bústia utilitzant un MUA i un mecanisme d'accés adequat (per exemple amb *thunderbird* usant el protocol POP o IMAP).

Exemple d'MDA

Podem trobar un exemple de lliurament(*delivery*) en el fitxer que indica els comptes de correu on reenviar els missatges que rep un usuari en un sistema Unix.

@ (at)

@ correspon al significat **at** en anglès o **a** en català. Usualment es diu usuari@màquina (usuari tal a la màquina qual), però no necessàriament el nom de domini correspon al nom de màquina. És més correcte dir usuari@domini.

Per exemple pere@gmail.com indica el compte de correu d'en Pere en la màquina gmail.com, però de fet no hi ha cap màquina que es digui així sinó que és el compte de correu d'en Pere al domini gmail.com. En realitat Gmail té varies màquines que responen a aquest nom de domini.

Correu web

El correu web té un funcionament similar al correu electrònic. Un usuari del Gmail utilitza com a MUA el web de Gmail per enviar un missatge a un usuari de Yahoo. El Gmail transfereix el missatge per SMTP al servidor de correu de Yahoo i el missatge es diposita en la bústia del destinatari. Aquest, quan li sembla, consulta el correu

1.1.1 El format dels missatges (RFC 822)

Exemples de programes
que implementen SMTP:
Sendmail, Exim, Postfix,
MS Exchange Server, etc.

El protocol SMTP s'encarrega del transport de missatges de correu amb independència del format i del contingut. Els missatges es componen de diferents elements que es descriuen entre l'especificació SMTP (corresponent al document RFC 821) i l'especificació pròpia dels missatges d'Internet (document RFC 822).

Podem desglossar el missatge en els elements següents:

- **Sobre o envelop.** Com passa en el correu postal, per fer arribar un missatge cal un sobre on s'indiqui el destinatari i el remitent. L'especificació d'SMTP (document RFC 821) descriu, com a sobre, el conjunt de dades necessàries per al transport del missatge (emissor, receptor, prioritat, nivell de seguretat, etc.). Generalment, el sobre consta únicament del camp FROM (emissor) i el camp RCPT (receptor). L'MTA utilitza el sobre per encaminar el missatge. De fet, la separació entre sobre i contingut és confusa i l'MTA obté les dades del sobre a partir de les capçaleres del contingut.
- **Contingut.** El contingut d'un missatge és el que està descrit en el document RFC 822 "Format d'un missatge". Tot contingut consta d'un conjunt de capçaleres o *headers*, una línia en blanc i un cos o *body* del missatge:

Capçaleres o headers. El contingut del missatge conté capçaleres del tipus *clau: valor* cada una en una línia independent.

Línia en blanc. Les capçaleres se separen del cos del missatge amb una línia en blanc.

Cos del missatge. El cos del missatge conté el missatge que es vol fer arribar al receptor. L'especificació inicial només permet text ASCII de 7 bits (sense símbols internacionals). El cos del missatge s'acaba amb una línia que conté a l'inici únicament un punt.

Les capçaleres descrites en l'especificació inicial tenen per objectiu poder descriure clarament l'emissor i el receptor o receptors del missatge, la data, permetre identificar el missatge (ID únic) entre d'altres. En l'exemple següent es poden veure els elements que formen el contingut i les capçaleres usals d'un correu electrònic.

Els camps FROM i RCPT
del sobre no porten dos
punts (:), i si que en
porten les capçaleres
FROM i RCPT del
contingut.

Són exemples de
capçaleres From:, To:,
Date:, Subject:, etc.

```
Received: by 10.100.195.12 with HTTP; Sun, 11 May 2008 10:11:38 -0700 (PDT)
Message-ID: <7b4e8fcc0805111011g83da6b0rdbd4f63409024720@gmail.com>
Date: Sun, 11 May 2008 19:11:38 +0200
From: "Pere Puig" <puig@correu.fp-oberta.org>
To: ppuig@correu.fp-oberta.org
Subject: =?ISO-8859-1?Q?Exemple_de_missatge_de_correu_amb_capçaleres
Delivered-To: ppuig@correu.fp-oberta.org

Hola,
Això és un exemple de missatge de correu
conté les capçaleres usals.
S'ha generat des de la web de gmail i s'envia també a gmail.

Pere
```

Aquestes són algunes de les capçaleres estàndard:

- **From.** Indica l'adreça de correu de l'emissor del missatge.
- **Sender.** Adreça de qui ha enviat el missatge. No s'utilitza si qui ha enviat el missatge és el mateix que el From. Serveix per diferenciar entre qui envia el missatge físicament i en nom de qui ho fa.
- **To i Cc.** Els camps To i Cc serveixen per indicar els destinataris del missatge. La idea original és posar un destinatari en el To i la resta en el Cc, però amb la utilització dels MUA actuals i la utilització de llistes d'usuaris, generalment, es posen tots els destinataris en el To.
- **Bcc.** Prové de *blind carbon copy* o còpia oculta. S'indiquen els destinataris que han de rebre el missatge però que no han d'aparèixer en la llista de destinataris. És per evitar que els altres destinataris sàpiguen qui n'ha rebut una còpia.
- **Reply-to.** Indica l'adreça de retorn del missatge al remitent. L'emissor pot voler que, si el missatge es retorna o es respon, l'adreça a la qual

es dirigeix la resposta sigui diferent de la indicada en el From. És útil per concentrar les respostes en un compte de correu quan l'emissor en té més d'un.

- **Received.** Cada MTA que processa un missatge afegeix una entrada de tipus *received* en el missatge. És una manera de realitzar el seguiment o traçabilitat dels MTA pels quals ha passat el missatge. La informació afegida descriu l'emissor (*from*) i el receptor (*by*), el mecanisme físic (*via*), l'identificador del missatge (*id*) i la data i hora (*date*).
- **Date.** Indica la data i hora en què s'ha generat el missatge. L'hi afegeix el primer MTA que rep el missatge del MUA.
- **Message-ID.** Identificador únic del missatge. Cada missatge s'ha de poder referenciar de manera única en tot el món. Això permet que les respostes indiquin a quin missatge es refereixen. S'utilitzen els noms de domini i un identificador numèric únic que genera l'MTA que rep el missatge per enviar.
- **Subject.** Descriu el propòsit del missatge o assumpte. És un petit text explicatiu.
- **In-reply-to.** Quan un missatge és una resposta a un missatge anterior, aquest camp indica a quin missatge original es fa referència.
- **Keywords.** Llista separada per comes de paraules clau descriptives del missatge.
- **Comments.** Text de comentari del missatge que no interfereix en el contingut.
- **References.** Quan un missatge fa referència a altres missatges anteriors, es poden indicar mitjançant aquesta capçalera.
- **Encrypted.** Indica el tipus de xifratge que s'ha utilitzat per xifrar el missatge. L'especificació del format dels missatges de correu (descrita en el document RFC 822) no indica cap tipus de xifratge, simplement reserva una capçalera per indicar-ne el tipus.
- **Return-path.** Identifica el camí de retorn cap a l'origen. Aquesta informació l'ha de posar l'MTA receptor i actualment està en desús, de manera que normalment conté l'adreça de l'emissor.
- **X-<userDefined>.** Els usuaris poden crear les pròpies capçaleres amb el nom que vulguin però començant per X-. D'aquesta manera s'assegura que si apareixen noves capçaleres oficials en el futur, no xocaran amb capçaleres definides pels usuaris (ja que aquestes han de començar totes per X-).

1.1.2. Funcionament del protocol SMTP.

El funcionament del protocol SMTP imita el correu postal en molts aspectes. L'SMTP és un protocol d'emmagatzemament i enviament igual que es fa amb les cartes de correu, que es lliuren a una oficina postal, d'allà a una altra i així fins a arribar al destinatari final. De fet, les cartes es lliuren a la bústia del destinatari final i és aquest el que les ha de recollir.

El **servidor SMTP** és una aplicació distribuïda que permet enviar missatges electrònics. Utilitza el protocol de transport TCP i el port 25.

En l'esquema original en què es va desenvolupar l'SMTP, una organització disposa d'un servidor SMTP (un MTA) que rep correu electrònic de fora de l'organització i el diposita en les bústies de correu locals del servidor. També recull el correu intern de l'organització i el canalitza per enviar-lo fora.

Cada organització disposa d'una o més màquines encarregades de gestionar el correu. Així, quan s'envia un correu electrònic a l'usuari *pere@ioc.cat*, cal que l'organització o domini *ioc.cat* disposi de màquines

Push

Es diu que l'SMTP és un protocol que fa push (lliurament) però no pull (agafar). Els usuaris finals han d'usar altres mecanismes per accedir remotament als seus comptes de correu.

MX

que fan la funció de servidors de correu. ¿Com trobarà l'SMTP a quin servidor de correu ha de lliurar els correus electrònics destinats a un domini? Utilitzant el protocol DNS (*domain name system*, sistema de noms de domini) obtindrà la màquina o màquines que fan la funció de servidors de correu del domini.

El client SMTP o emissor estableix una connexió TCP amb el port 25 del servidor SMTP o receptor. En una mateixa connexió l'emissor pot enviar un o més missatges al receptor. Si el mateix missatge va destinat a diversos receptors del sistema final, el missatge s'envia un sol cop i l'MTA receptor ja l'expandirà per a cada destinatari.

El client SMTP o emissor disposa d'una cua de missatges per enviar i una llista de destinataris per a cada missatge. Els destinataris poden ser en destinacions diferents (evidentment) i, per tant, li caldrà connectar-se als diferents servidors de destinació per tal de fer-los arribar els missatges.

Quan un destinatari no és accessible, el missatge es pot tornar a posar en la cua de missatges pendents d'enviar o es pot descartar (segurament després de diferents intents infructuosos) tot intentant notificar l'emissor.

Avui en dia el servidor de correu pot ser a qualsevol lloc del món i no cal que cada organització en tingui d'un. Es pot utilitzar el del proveïdor ISP o el de qualsevol servei extern de correu (per exemple, Google ofereix el servei d'externalitzar el correu a empreses tot mantenint el domini propi de l'empresa). Això significa que el servidor SMTP ha de verificar si accepta o no peticions d'enviar correu d'un client. Es pot verificar el client mitjançant l'IP o mitjançant altres mecanismes d'autenticació i seguretat. Evidentment, disposar d'un servidor SMTP que accepta peticions de clients sense verificar qui són és una porta oberta a permetre correu brossa. Normalment els servidors SMTP restringeixen qui pot fer ús del servei (quins clients) i a quines destinacions.

Un cop un servidor SMTP accepta un correu electrònic per fer-ne el lliurament (d'un MUA com el thunderbird per exemple) tot validant que accepta rebre correus electrònics d'aquest client, estableix una connexió TCP al port 25 del servidor SMTP destinatari (ha obtingut l'adreça IP fent la resolució DNS de la part del domini de l'adreça de correu).

En la base de dades d'un servidor DNS, els equips que fan de servidors de correu d'un domini s'identifiquen per les entrades tipus MX. Si un domini no disposa d'entrades MX s'utilitza el *host* que defineix el domini.

Listes negres de servidors de correu

Els servidors de correu que accepten tot tipus de correus electrònics de tots els clients a totes les destinacions es posen en llistes negres perquè poden ser generadors de correu brossa.

Correu brossa o spam

S'anomena *correu brossa* el correu no desitjat o no sol·licitat. És un correu que es rep insistentment i que bombardeja les bústies dels usuaris de manera mecànica.

1.1.3.Ordres/Respostes SMTP.

L'emissor sempre porta el control de la comunicació i inicia la connexió amb el receptor. El diàleg consisteix en un intercanvi d'ordres i respostes que segueixen les especificacions de *Telnet*:

- **Ordres.** Les ordres són codis de quatre caràcters i arguments opcionals separats per espais i acabats amb <CRLF> (HELO, MAIL, DATA, etc.). Per a cada ordre es rep una resposta del receptor.
- **Respostes.** Les respostes són codis numèrics de tres dígits, un espai i un missatge descriptiu que pot variar segons la implementació.

Un diàleg bàsic entre emissor i receptor SMTP seria el següent:

- **HELO domini / EHLO domini.** Un cop connectat, l'emissor s'ha d'identificar amb l'ordre HELO i indicar el domini al qual es connecta. Actualment els servidors SMTP utilitzen extensions i l'ordre preferida per identificar-se és EHLO (significa extended HELO).
- **MAIL FROM: <emissor>.** Identifica l'emissor del missatge i genera la capçalera FROM del missatge. El receptor valida que l'emissor sigui un usuari vàlid, és a dir, que accepti missatges d'aquest origen. Si no el pot validar envia una resposta denegant-ho. Els equips amb el *relay* configurat per permetre enviar missatges de tothom són els principals generadors de correu brossa.
- **RCPT TO <destinatari>.** Indica el destinatari del missatge. Aquesta ordre es pot repetir tantes vegades com destinataris tingui el missatge. També cal que el receptor accepti el destinatari, que pot ser un destinatari local, o que accepti fer el reenviament si és un destinatari remot. Aquesta ordre genera la capçalera TO en el missatge.
- **DATA.** Indica que a continuació s'enviarà el missatge. Tot el que es transmet a continuació és el contingut del missatge, que finalitzarà en

S'entén per CRLF una línia en blanc.

trobar una línia que només inclou un punt (<CRLF>). El contingut segueix les especificacions del document RFC 822, per tant, pot contenir capçaleres a l'inici, una línia en blanc a manera de separador i el cos. No es pot enviar un missatge (l'ordre DATA) fins que el receptor no ha confirmat que accepta almenys un destinatari. Això evita transmetre missatges que es descartarien en la destinació.

- **QUIT.** L'emissor envia l'ordre per indicar al receptor que vol finalitzar la comunicació. El receptor confirma la recepció i llavors tots dos poden finalitzar la transmissió.

En l'exemple següent podeu veure un diàleg client/servidor SMTP mitjançant ordres i respostes Telnet.

```
[root@host ~]# telnet www.escola.org 25
Trying 22.170.21.168...
Connected to www.escola.org.
Escape character is '^]'.
220 escola.org ESMTP Sendmail 8.13.8/8.13.8; Sat, 26 Apr 2008
19:56:05 +0200
EHL0 escola.org
250-escola.org Hello 106.Red-71-92-14.dynamicIP.rima-tde.net
71.92.14.106], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 10000000
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250-DELIVERBY
250 HELP

MAIL FROM: pere@xtec.cat
250 2.1.0 pere@xtec.cat... Sender ok
RCPT TO: pere@correu.escola.org
250 2.1.5 pere@correu.escola.org... Recipient ok

DATA
354 Enter mail, end with "." on a line by itself
hola,
aquest és un missatge de prova per enviar un
email usant telnet al servidor smtp de l'escola.
s'envia una còpia a dos usuaris locals al servidor.
S'ha denegat fer relaying i enviar una còpia a
l'exterior
pere
.
250 2.0.0 m3QH5B3012660 Message accepted for delivery

QUIT
221 2.0.0 escola.org closing connection
Connection closed by foreign host.
```

Amb els camps MAIL FROM i RCPT TO, el protocol SMTP obté les dades necessàries per generar el sobre o *envelop*.

A part de les ordres bàsiques mostrades anteriorment hi ha més ordres en el protocol SMTP, com ara les següents:

- **RSET.** L'emissor pot interrompre l'enviament de missatges amb aquesta ordre.
- **NOOP.** Aquesta ordre no fa res, però força el receptor a enviar una resposta afirmativa. Serveix com a mecanisme per confirmar que la connexió encara és oberta.
- **HELP.** Fa una llista de les ordres que implementa el servidor. Els servidors SMTP no implementen necessàriament totes les ordres descrites pel protocol.
- **VRFY <destinatari>.** L'emissor pot verificar l'existència del destinatari.
- **EXPN <destinatari>.** Permetrà a l'emissor verificar l'existència d'una llista de correu i obtenir-ne la llista dels membres.
- **SEND, SOML, SAML.** Aquestes ordres permeten enviar els missatges tant a les bústies de correu com als terminals.
- **TURN.** Permet intercanviar els papers entre emissor i receptor. El receptor hi ha d'estar d'acord.

Els servidors SMTP no implementen necessàriament totes les ordres, hi ha un conjunt d'ordres mínim definit pel protocol que tot servidor SMTP ha d'implementar.

El conjunt d'ordres mínim que ha d'implementar un servidor SMTP ha de ser el següent:

Ordres SMTP

Per obtenir la llista d'ordres del protocol podeu consultar el document RFC 2821. Atès un servidor concret, podeu consultar les ordres que implementa amb l'ordre HELP.

HELO <domini> MAIL FROM: <emissor> RCPT TO: <destinatari>
DATA RSET NOOP QUIT

El protocol SMTP permet treballar amb missatges ASCII de 8 bits i amb extensions del protocol, és a dir, afegir als servidors SMTP funcionalitats extres segons el programari de servidor utilitzat. El client pot sol·licitar al receptor la llista de les extensions que implementa i fer-li saber que les vol utilitzar. El mecanisme consisteix en que el client envii un **EHLO <domini>** en lloc de l'HELO estàndard. Si el receptor implementa extensions respondrà afirmativament i en farà una llista, si no les implementa respondrà negativament. Llavors l'emissor pot fer un HELO estàndard.

Les respostes es poden classificar en quatre grans grups. El primer dígit del codi numèric de tres dígits de la resposta indica al grup al qual pertany:

1)Positiva (2) . L'acció que ha sol·licitat l'emissor és acceptada pel receptor. L'emissor pot fer una nova sol·licitud.

2)Intermèdia positiva (3) . L'acció sol·licitada s'ha acceptat però està suspesa pendent de rebre informació addicional que l'emissor haurà de proporcionar.

3)Negativa transitòria (4) . La sol·licitud no s'ha acceptat i l'acció no s'ha realitzat, però es tracta d'un error temporal i es pot tornar a intentar més tard. L'emissor pot tornar a fer la sol·licitud més endavant.

4)Negativa pertinent (5) . L'ordre no s'ha realitzat i, per tant, la sol·licitud no ha estat acceptada.

1.1.4.Els tipus MIME.

Els missatges de correu segueixen el format definit en el document RFC 822 (actualment, RFC 2822), que únicament permet missatges de text net ASCII de 7 bits. No es permeten els caràcters accentuats, caràcters internacionals (ASCII de 8 bits) i molt menys la transferència de dades binàries com imatges, àudio, aplicacions, PDF, etc. Però tot això i molt més s'envia avui en dia per correu electrònic, com?

El juny del 1992 es va definir el que es coneix com a **MIME(multipurpose Internet mail extension o extensió de correu d'Internet per a ús múltiple)** en l'RFC 1341, i que actualment ha evolucionat en els RFC 2045 i RFC 2049. El MIME utilitza missatges RFC 822 però afegint una estructura al cos del missatge i regles de codificació per a missatges no ASCII. El gran avantatge del MIME és que permet seguir utilitzant les mateixes eines d'SMTP que fins ara, només cal modificar els MUA perquè apliquin MIME. A l'MTA el cos del missatge li és absolutament indiferent (i per tant pot estar codificat), només utilitza el sobre per enviar el missatge i el contingut s'envia com un tot.

El MIME es basa en tres elements per permetre qualsevol tipus de contingut en un missatge de correu:

- **Capçaleres MIME.** Es creen cinc noves capçaleres de correu per definir informació del cos del missatge. No totes són obligatòries.
- **Formats de contingut.** Es defineixen diferents formats de contingut que permeten als MUA receptors interpretar el contingut de manera adequada i saber si reben un full de càlcul, un vídeo, etc.
- **Esquemes de codificació de transferència.** Es realitza una transformació de les dades a un format manipulable per al transport SMTP (que només permet caràcters ASCII de 7 bits).

Podeu veure els components d'un missatge amb contingut MIME en l'exemple següent.

```
From root@tftp.server.cat Fri Jun 13 17:26:31 2008
Return-Path: <root@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
  by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DFQTH7003922
  for <pere@tftp.server.cat>; Fri, 13 Jun 2008 17:26:30 +0200
Received: (from root@localhost)
  by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DFQSIq003918
  for pere@tftp.server.cat; Fri, 13 Jun 2008 17:26:28 +0200
```



```

Date: Fri, 13 Jun 2008 17:26:27 +0200
From: root <root@tftp.server.cat>
To: pere@tftp.server.cat
Subject: missatge amb atachment
Message-ID: <20080613152627.GA3909@portatil.local.lan>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="zYM0uCDKw75PZbzx"
Content-Disposition: inline
User-Agent: Mutt/1.5.17 (2007-11-01)
Status: 0

--zYM0uCDKw75PZbzx
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline

misatge de root a l'usuari pere
conte adjunt un pdf i jpeg
adeu!

--zYM0uCDKw75PZbzx
Content-Type: application/pdf
Content-Disposition: attachment;
filename="informatica_AX_ud2.pdf"
Content-Transfer-Encoding: base64
... output suprimir (contingut del pdf codificat en base64) ...
--zYM0uCDKw75PZbzx
Content-Type: image/jpeg
Content-Disposition: attachment; filename="cd15_11_puerto-
madrin.jpg"
Content-Transfer-Encoding: base64
... output suprimir (contingut del jpeg codificat en base64) ...
--zYM0uCDKw75PZbzx--

```

Capçaleres MIME

Les cinc capçaleres que defineix l'especificació MIME aporten informació del contingut del missatge. Aquestes capçaleres són les següents:

- **MIME-version.** Identifica el tipus MIME del missatge. Si indica 1.0 es tracta d'un missatge MIME, si no, es tracta d'un missatge ASCII.
- **Content-description.** És un text que descriu el tipus de contingut. No és obligatori i no té cap funcionalitat més enllà de la merament descriptiva.
- **Content-ID.** Identifica el contingut de manera única igual que ho fa el camp Message-ID.
- **Content-transfer-encoding.** Mecanisme de codificació utilitzat en el missatge per poder-lo transmetre. El contingut que no és ASCII de 7 bits es codifica per poder ser transmès.
- **Content-type.** Descriu el tipus de contingut segons la taula de *MIME types*. Això permet a un MUA obrir l'aplicació pertinent per gestionar el contingut. Si, per exemple, el tipus és *image/jpeg* permet al MUA saber que en pot manipular el contingut amb una aplicació de gestió d'imatges.

Tipus MIME

Es defineix un conjunt de tipus i subtipus MIME amb un esquema tipus/subtipus. Originàriament es van definir els set tipus que es descriuen a continuació, però en l'actualitat hi ha multitud de tipus/subtipus MIME:

- **text/native.** Text net en format ASCII de 7 bits.
- **multipart/<subtipus>.** El missatge conté múltiples parts independents. Un delimitador (o *boundary*) indica la separació de cada part. El delimitador és únic i no apareix en el cos de les parts. El delimitador es troba a l'inici i al final de cada part i comença amb dos guions. L'última part acaba amb un delimitador que comença amb dos guions i acaba amb dos guions. Cada part pot ser qualsevol cosa!
- **multipart/parallel.** Múltiples parts, en ordre. És a dir, les parts s'han de mostrar en el receptor en l'ordre indicat.
- **multipart/mixed.** Múltiples parts. No es defineix cap ordre.
- **multipart/alternative.** Les parts són versions alternatives del mateix

contingut en ordre creixent de fidelitat. El receptor escull la més apropiada. Per exemple, un text s'envia com a text pla en una primera part i com a PDF en una segona; si el receptor no disposa de PDF podrà usar la part en text net.

- **multipart/digest.** Cada part és un missatge de correu individual. S'utilitza quan un correu electrònic conté diferents correus electrònics en a l'interior (per exemple, reenviaments).
- **message/rfc822.** El cos és un missatge de correu complet, amb capçaleres i cos. Pot ser un missatge MIME tot i que el nom digui rfc822.
- **message/partial.** Permet fragmentar un missatge llarg en diferents missatges. Cada fragment ha de disposar d'un identificador, número de fragment i nombre total de fragments.
- **message/external body.** Les dades del cos del missatge no estan en el missatge sinó que cal baixar-les a part. En la capçalera *Content-Type* es descriu el tipus de contingut i el tipus d'accés, que pot ser FTP, TFTP, anon-FTP (FTP anònim), *local-file*, AFI i *mail-server*. Per exemple, el contingut pot ser una imatge no inclosa en el missatge sinó que cal baixar-lo d'un servidor FTP.
- **image/jpeg.** Imatge codificada JPEG.
- **image/gif.** Imatge GIF.
- **video/mpeg.** Vídeo en format MPEG (*moving picture experts group*, grup d'experts d'imatges en moviment).
- **audio/basic.** Àudio en format estàndard.
- **application/postscript.** Dades binàries en format PostScript, per exemple, PDF.
- **application/octet-stream.** Dades binàries.

Codificació de transferència

Les dades binàries i els caràcters internacionals (que no pertanyen al conjunt ASCII de 7 bits) no es poden enviar per correu electrònic. Per poder-ho fer cal codificar-los en un altre format. L'especificació MIME defineix els tipus de codificacions següents:

- **7bit.** Indica que les dades es transfereixen en ASCII de 7 bits. No es realitza cap codificació.
- **8bit.** No es realitza cap codificació i les dades es transmeten en ASCII de 8 bits. Evidentment, cal que receptor i emissor permetin la transferència a 8 bits (una extensió d'SMTP).
- **Binary.** Es transmeten les dades en binari tal com són, sense cap codificació ni control de la longitud de les línies. Si s'envien dades en binari (en cru) no es garanteix que la transmissió sigui correcta.
- **X-token.** Indica la utilització d'un esquema de codificació de transport no estàndard, un esquema propi. Emissor i receptor han de compartir aquest esquema de codificació.
- **Quoted-printable.** Quan la majoria de caràcters del missatge són imprimibles excepte uns pocs, és més eficient utilitzar aquesta codificació que base64. Aquest esquema codifica els caràcters no imprimibles amb un igual i el codi hexadecimal del caràcter. Es garanteix que les línies tenen una longitud no superior a setanta-sis caràcters mitjançant salts de línia reversibles.
- **Base64.** És l'esquema de codificació més usat per a la transferència d'informació binària. Converteix l'entrada en un conjunt de caràcters imprimibles i, per tant, immunes al transport per SMTP. Consta d'un conjunt de seixanta-tres caràcters imprimibles i un més de farciment ($2^6 = 64$ caràcters). Cada 24 bits de l'entrada binària (3 bytes) es codifica en quatre blocs de 6 bits ($4 \times 6 = 24$ bits). A cada bloc de 6 bits correspon un caràcter imprimible que es posa en 1 byte. Per tant, per cada 24 bits d'entrada binària, s'utilitzen 32 bits de transmissió (4×8

!!

Per aprendre el funcionament de "base64" podeu consultar la wikipèdia:
<http://en.wikipedia.org/wiki/Base64>
[<http://en.wikipedia.org/wiki/Base64>]

byte).

Base64:

Aquest és un petit exemple extret de la wikipèdia, on s'observa que el text "Man" original (3 bytes = 24 bits) acaba codificat en base64 com a "TWFu" (4 bytes).

```
Text content M a n
ASCII 77 97 110
Bit pattern 01001101 01100001 01101110 (8 bits x byte)
Bit ppattern 010011 010110 000101 101110 (divisió en blocs de 6 bits)
Index 19 22 5 46
Base64-encoded T W F u
```

1.2. Instal·la i configura un servidor de correu electrònic.

Hi ha diverses aplicacions de servidor de correu en el mercat i moltíssims clients de correu de tot tipus. Existeixen tant versions gràfiques com aplicacions en entorn de text. Algunes d'aquestes aplicacions són d'ús públic i que es poden baixar gratuïtament d'internet.

En els sistemes GNU/Linux la majoria d'ells proporcionen l'aplicació client **mail** i sovint també **mutt**, que és una versió de mail amb pantalletes en mode text. També disposen d'una aplicació servidor omnipresent en tots els sistemes GNU/Linux i Unix anomenada **sendmail**.

Així doncs, quan parlem d'instal·lar el servei de correu fem referència al procés d'instal·lació i configuració del programari del servidor. Això es fa de manera molt similar a la d'altres serveis de xarxa (com els serveis DHCP, DNS, HTTP, FTP, etc), es tracta d'instal·lar els paquets o *tarballs* de l'aplicació servidor i fer-ne la configuració apropiada. Senzill oi?

Per fer això cal plantejar-se els següents passos:

- Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
- Observar l'estat de la xarxa actual. Està el servei ja en funcionament? Existeix ja un servidor de correu instal·lat i actiu?
- Obtenir l'aplicació que proporciona el servei de correu.
- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha efectuat correctament.
- Configurar el servei en el servidor i provar que els clients hi poden accedir.
- Comprovar que el servei funciona correctament.

Usualment l'administrador acaba utilitzant l'aplicació servidor que li proporciona el propi sistema operatiu que està utilitzant. Si utilitzeu el sistema operatiu Windows l'empresa Microsoft disposa d'una aplicació pròpia, però també en podeu trobar d'altres a Internet. Igualment si utilitzeu GNU/Linux segurament la mateixa distribució ja proporciona un servidor de correu o bé n'existeix algun de clàssic provinent de l'*Unix*. De totes maneres en podeu obtenir d'altres també a Internet.

L'eina que s'utilitzarà en aquest dossier per oferir els serveis de servidor de correu és **sendmail**. Podeu trobar tota l'informació d'aquest servidor a: **www.sendmail.org**.

1.2.1. Instal·lació de l'aplicació servidor.

Tot seguit es descriurà el procés per instal·lar el servei de correu en un entorn GNU/Linux. Un cop feta la instal·lació cal observar què s'ha instal·lat, quins programes executables, on són els fitxers de configuració, els de monitoratge, etc.

Els usuaris GNU/Linux poden buscar fàcilment per Internet quins paquets de servidor de correu *sendmail* hi ha usant eines com *yum* o *apt-get* i els repositoris de paquets apropiats segons quina sigui la distribució que utilitzin. A més a més sempre hi ha l'omnipresent eina *Google* per ajudar a localitzar tot allò que faci falta.

Aplicacions de servidor de correu recomanables

Més que fer una enumeració de les aplicacions de correu actuals, us suggerim que consulteu per Internet quines aplicacions estan "de moda" i són les recomanades per a cada tipus de sistema operatiu. Esbrineu les característiques que diferencien les unes de les altres.

Cerca de *sendmail* a Internet

Usualment l'administrador s'informa a través del seu cercador preferit, per exemple *Google*, i de webs com la *wiki* (*viquipèdia*). Proveu a buscar "sendmail" o "SMTP server" al *Google* i a la *wiki*.

Per fer 'cultura' sobre els orígens i l'evolució de *sendmail* es recomana consultar la informació que es mostra a la *wikipèdia*.

Un cop instal·lat el software caldrà identificar què s'ha instal·lat. Quins paquets i què contenen. A vegades no s'instal·laran paquets sinó fitxers *tarball* dels que també caldrà saber examinar-ne el contingut. És important identificar quins dels components instal·lats corresponen a fitxers executables, a fitxers de configuració i a fitxers de documentació.

Tot servei instal·lat s'ha de configurar apropiadament i posar en marxa. Per tant caldrà saber gestionar l'estat del servei (engegar, aturar, recarregar, etc) i definir l'estat que ha de tenir en els diferents *runlevels* del sistema.

En definitiva, el procediment d'instal·lar inclourà usualment:

- buscar el software del servei (sigui en format de paquets *.deb*, *.rpm* o paquets *.tar*) i descarregar-lo utilitzant l'eina apropiada segons sigui la distribució.
- examinar el sistema per identificar quin software, quins paquets, hi ha instal·lats relacionats amb el servei.
- identificar els components del servei. Quins són els fitxers executables, de configuració i de documentació.
- consultar i establir l'estat del servei (engegar i aturar) i saber establir l'estat per defecte per a cada *runlevel*.

Instal·lació en un sistema Fedora

Els següents fragments de codi mostren tot el procés per identificar, instal·lar i examinar els paquets del servidor de correu *sendmail* usant un GNU/Linux Fedora.

Llistar els paquets que contenen la cadena *sendmail* i instal·lar el paquet corresponent al servidor. Usualment no caldrà realitzar aquesta tasca perquè el servei està instal·lat en tots els sistemes. Si volem comprovar si el sistema ja té instal·lats aquests paquets podem consultar quins paquets *sendmail* hi ha actualment instal·lats fent:

```
[root@host ~]# yum search sendmail
# Instal·lar el paquet servidor
[root@host ~]# yum install sendmail

# Observar quins paquets instal·lats hi ha amb la cadena sendmail
[root@host ~]# rpm -qa | grep sendmail
sendmail-8.14.1-4.2.fc7
```

Obtenir informació del paquet del servei *sendmail*:

```
[root@host ~]# rpm -qi sendmail
Name       : sendmail                      Relocations: (not relocatable)
Version    : 8.14.1                      Vendor: Fedora Project
Release    : 4.2.fc7                     Build Date: dl 17 set 2007 19:49:22 CEST
Install Date: dc 26 set 2007 20:35:07 CEST Build Host: xenbuilder4.fedora.phx.redhat.com
Group      : System Environment/Daemons  Source RPM: sendmail-8.14.1-4.2.fc7.src.rpm
Size       : 1933720                     License: Sendmail
Signature  : DSA/SHA1, dt 18 set 2007 22:13:30 CEST, Key ID b44269d04f2a6fd2
Packager   : Fedora Project
URL        : http://www.sendmail.org/
Summary    : A widely used Mail Transport Agent (MTA)
Description:
The Sendmail program is a very widely used Mail Transport Agent (MTA).
MTAs send mail from one machine to another. Sendmail is not a client
program, which you use to read your email. Sendmail is a
behind-the-scenes program which actually moves your email over
networks or the Internet to where you want it to go.

If you ever need to reconfigure Sendmail, you will also need to have
the sendmail-cf package installed. If you need documentation on
Sendmail, you can install the sendmail-doc package.
```

Observar els components del paquet

Fer la llista dels components del paquet *sendmail*:

```
[root@host ~]# rpm -ql sendmail
/etc/mail
/etc/mail/Makefile
/etc/mail/access
/etc/mail/access.db
/etc/mail/domaintable
/etc/mail/domaintable.db
/etc/mail/helpfile
/etc/mail/local-host-names
```

```

/etc/mail/mailertable
/etc/mail/mailertable.db
/etc/mail/sendmail.cf
/etc/mail/sendmail.mc
/etc/mail/submit.cf
/etc/mail/submit.mc
/etc/mail/trusted-users
/etc/mail/virtusertable
/etc/mail/virtusertable.db
/etc/pam.d/smtp.sendmail
/etc/rc.d/init.d/sendmail
/etc/smrsh
/etc/sysconfig/sendmail
/usr/bin/hoststat
/usr/bin/mailq.sendmail
/usr/bin/makemap
/usr/bin/newaliases.sendmail
/usr/bin/purgestat
/usr/bin/rmail.sendmail
/usr/lib/sasl2/Sendmail.conf
/usr/lib/sendmail.sendmail
/usr/sbin/mailstats
/usr/sbin/makemap
/usr/sbin/praliases
/usr/sbin/sendmail.sendmail
/usr/sbin/smrsh
/usr/share/doc/sendmail-8.14.1
... output suprimit ...
/usr/share/man/man8/smrsh.8.gz
/var/log/mail
/var/log/mail/statistics
/var/spool/clientmqueue
/var/spool/mqueue

```

En funció del directori on s'ubiquen els fitxers podem intuir si són executables, de configuració o de documentació. També podem mirar de filtrar la sortida en cada cas.

Fitxers de configuració:

```

[root@host ~]# rpm -qlc sendmail
/etc/mail/Makefile
/etc/mail/access
/etc/mail/daemons
/etc/mail/helpfile
/etc/mail/local-host-names
/etc/mail/mailertable
/etc/mail/sendmail.cf
/etc/mail/sendmail.mc
/etc/mail/submit.cf
/etc/mail/submit.mc
/etc/mail/trusted-users
/etc/mail/virtusertable
/etc/pam.d/smtp.sendmail
/etc/sysconfig/sendmail
/usr/lib/sasl2/Sendmail.conf
/var/log/mail/statistics

[root@host ~]# rpm -ql sendmail | grep etc
/etc/mail
/etc/mail/Makefile
/etc/mail/access
/etc/mail/access.db
/etc/mail/daemons
/etc/mail/daemons.db
/etc/mail/helpfile
/etc/mail/local-host-names
/etc/mail/mailertable
/etc/mail/mailertable.db
/etc/mail/sendmail.cf
/etc/mail/sendmail.mc
/etc/mail/submit.cf
/etc/mail/submit.mc
/etc/mail/trusted-users
/etc/mail/virtusertable
/etc/mail/virtusertable.db
/etc/pam.d/smtp.sendmail
/etc/rc.d/init.d/sendmail
/etc/smrsh
/etc/sysconfig/sendmail

```

Fitxers de documentació:

```

[root@host ~]# rpm -qld sendmail
/usr/share/doc/sendmail-8.14.4/FAQ
/usr/share/doc/sendmail-8.14.4/KNOWNBUGS
/usr/share/doc/sendmail-8.14.4/LICENSE
/usr/share/doc/sendmail-8.14.4/README
/usr/share/doc/sendmail-8.14.4/RELEASE_NOTES.gz
/usr/share/man/man1/mailq.sendmail.1.gz
/usr/share/man/man1/newaliases.sendmail.1.gz
/usr/share/man/man5/aliases.sendmail.5.gz
/usr/share/man/man8/mailstats.8.gz
/usr/share/man/man8/makemap.8.gz
/usr/share/man/man8/praliases.8.gz
/usr/share/man/man8/rmail.8.gz
/usr/share/man/man8/sendmail.sendmail.8.gz
/usr/share/man/man8/smrsh.8.gz

```

Podem mirar de filtrar quins són els executables tenint en compte que usualment estaran en un directori de nom *bin* o *sbin*. En aquest cas hi ha un únic executable corresponent al dimoni del servei.

```
[root@host ~]# rpm -ql sendmail | grep "bin"
/usr/bin/hoststat
/usr/bin/mailq.sendmail
/usr/bin/makemap
/usr/bin/newaliases.sendmail
/usr/bin/purgestat
/usr/bin/rmail.sendmail
/usr/sbin/mailstats
/usr/sbin/makemap
/usr/sbin/praliases
/usr/sbin/sendmail.sendmail
/usr/sbin/smrsh
```

Finalment sempre és aconsellable observar l'estructura de directoris i fitxers que ha generat la instal·lació del paquet:

```
# locate sendmail
/etc/NetworkManager/dispatcher.d/10-sendmail
/etc/alternatives/mta-sendmail
/etc/alternatives/mta-sendmailman
/etc/mail/sendmail.cf
/etc/mail/sendmail.mc
/etc/pam.d/smtp.sendmail
/etc/rc.d/init.d/sendmail
/etc/rc.d/rc0.d/K30sendmail
/etc/rc.d/rc1.d/K30sendmail
/etc/rc.d/rc2.d/S80sendmail
/etc/rc.d/rc3.d/S80sendmail
/etc/rc.d/rc4.d/S80sendmail
/etc/rc.d/rc5.d/S80sendmail
/etc/rc.d/rc6.d/K30sendmail
/etc/sysconfig/sendmail
/usr/bin/mailq.sendmail
/usr/bin/newaliases.sendmail
/usr/bin/rmail.sendmail
/usr/lib/sendmail
/usr/lib/sendmail.sendmail
/usr/lib/evolution-data-server-1.2/camel-providers/libcamelsendmail.so
/usr/lib/evolution-data-server-1.2/camel-providers/libcamelsendmail.urls
/usr/lib/python2.6/site-packages/sos/plugins/sendmail.py
... output suprimir ...
/var/lock/subsys/sendmail
/var/run/sendmail.pid
```

En resum:

- Els fitxers de documentació es troben generalment a: */usr/share/doc* i a */usr/share/man*.
- Els fitxers de configuració es troben a: */etc/mail*.
- El dimoni del servei es troba a: */usr/sbin* i s'anomena *sendmail.sendmail*.
- El fitxer de configuració del dimoni del servei són els fitxers: */etc/mail/sendmail.mc* i */etc/mail/sendmail.cf*. El fitxer de configuració del servei és el *sendmail.cf*, però és un fitxer que només els valents i autèntics experts s'atreveixen a tocar. Generalment la configuració es defineix en el fitxer *sendmail.mc* i es processa amb *m4*.
- El fitxer de govern del servei és: */etc/rc.d/init.d/sendmail*.
- El fitxer de configuració del registre de *logs* es troba a */var/log/mail/statistics*.
- Es pot observar que la configuració de l'autenticació d'usuaris contra el sistema utilitzant PAM es governa a través d'un fitxer específic del servei de correu per el PAM anomenat *etc/pam.d/smtp.sendmail*.
- El directori de gestió de cues de missatges: */var/spool/clientmqueue* i */var/spool/mqueue*.

Ubicació de fitxers

En GNU/Linux els fitxers executables per l'administrador es troben usualment a */sbin* i a */usr/sbin*. Els executables d'usuari normalment són a */bin* i */usr/bin*.

```
# Eliminars els fitxers de 'doc' i 'man' el paquet consta de:
# rpm -ql sendmail | grep -v "doc" | grep -v "man"
... output suprimir ...
/etc/mail/aliasesdb-stamp
/etc/mail/sendmail.cf
/etc/mail/sendmail.mc
/etc/mail/submit.cf
/etc/mail/submit.mc
/etc/pam.d/smtp.sendmail
/etc/rc.d/init.d/sendmail
/etc/smrsh
/etc/sysconfig/sendmail
/usr/bin/hoststat
/usr/bin/mailq.sendmail
/usr/bin/makemap
/usr/bin/newaliases.sendmail
/usr/bin/purgestat
/usr/bin/rmail.sendmail
... output suprimir ...
/usr/sbin/mailstats
/usr/sbin/makemap
/usr/sbin/praliases
/usr/sbin/sendmail.sendmail
/usr/sbin/smrsh
/var/log/mail
/var/log/mail/statistics
/var/spool/clientmqueue
/var/spool/mqueue
```

Activar/desactivar el servei i establir els nivells d'arrencada

Un cop s'ha instal·lat al sistema un servei de xarxa, cal posar-lo en funcionament. Primer caldrà determinar quin tipus de servei és: si autònom o integrat dins del superservei de xarxa. Un cop fet això, cal saber si ja està en funcionament o no. De fet cal saber engegar-lo, aturar-lo i reinicialitzar-lo. Finalment, cal establir quin estat ha de tenir el servei per defecte cada cop que s'engegui el servidor.

• El servei

Primerament cal saber si el servidor instal·lat funciona “*stand-alone*” o dins del superdimoni de xarxa “*xinetd*” o “*initd*”. Si existeixen fitxers de configuració dins del directori */etc/xinetd.d/<nom-servei>* es tracta d'un servei dins del *xinetd*. Si existeixen fitxers de configuració dins del directori */etc/rc.d/init.d/<nom-servei>* es tracta d'un servei “*stand-alone*”.

Observem ara el contingut del paquet per intentar saber si els fitxers de configuració ens permeten saber de quin tipus de servei es tracta:

```
[root@host ~]# rpm -ql sendmail | grep etc
...
/etc/rc.d/init.d/sendmail
...
```

Com podem observar es tracta d'un servei *stand-alone*. També es pot consultar el tipus de servei amb l'ordre *chkconfig* i observar si surt la llista d'un tipus o de l'altre:

```
[root@host ~]# chkconfig --list sendmail
sendmail      0:aturat      1:aturat      2:executant-se 3:executant-se
4:executant-se 5:executant-se 6:aturat
```

• Estat del servei

Es pot saber l'estat del servei amb l'opció *status*, l'opcions *start* engega el servei, l'opció *stop* atura el servei, l'opció *reload* o *restart* l'atura i el reinicialitza. Per saber les opcions de govern possibles del servei es pot cridar sense cap argument. En sistemes Fedora es pot usar l'ordre *service* que de fet és una 'drecera' per indicar la ruta apropiada “*/etc/init.d/<nom-servei>*”.

```
[root@host ~]# service sendmail status
sendmail      està aturat
[root@host ~]# /etc/rc.d/init.d/sendmail status
sendmail      està aturat

[root@host ~]# /etc/rc.d/init.d/sendmail start
S'està iniciant el servei sendmail:      [ FET ]

[root@host ~]# service sendmail stop
S'està aturant el sendmail:              [ FET ]

[root@host ~]# /etc/rc.d/init.d/sendmail restart
S'està aturant el servei sendmail:      [Incorrecte]
S'està iniciant el servei sendmail:      [ FET ]

[root@host ~]# service sendmail reload
S'està actualitzant el sendmail:         [ FET ]

[root@host ~]# service sendmail
Ús: /etc/init.d/sendmail {start|stop|restart|condrestart|status}
```

• Establir els nivells per defecte del servei

Els serveis (els dimonis executables) es poden configurar per arrencar automàticament en determinats nivells d'execució. Les màquines GNU/Linux tenen 7 nivells d'execució com es pot veure del fitxer */etc/inittab*:

```
[root@host ~]# head -20 /etc/inittab
... output suprimir ...
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
# ... output suprimir ...
```

Per configurar a quins nivells es vol que s'executi un servei s'utilitza l'ordre **chkconfig** que permet activar/desactivar el servei pels nivells indicats. Podem usar-la també per llistar l'estat per defecte del servei per a cada *runlevel*.

```
[root@host ~]# chkconfig --list sendmail
```

```

sendmail      0:aturat      1:aturat      2:executant-se 3:executant-se
4:executant-se 5:executant-se 6:aturat

[root@host ~]# chkconfig --help
chkconfig versió 1.3.34 - Copyright (C) 1997-2000 Red Hat, Inc.
Aquest programari es pot distribuir lliurement d'acord amb els termes de la Llicència Pública
General GNU.
forma d'ús:  chkconfig --list [nom]
             chkconfig --add <nom>
             chkconfig --del <nom>
             chkconfig --override <nom>
             chkconfig [--level <nivells>] <nom> <on|off|reset|resetpriorities>

[root@host ~]# chkconfig --level 345 sendmail off
[root@host ~]# chkconfig --list sendmail
sendmail      0:aturat      1:aturat      2:executant-se 3:aturat
4:aturat      5:aturat      6:aturat

```

Definir els nivells d'execució amb l'ordre *chkconfig* no significa que el servei s'engegui en aquest instant, sinó que significa que quan arranqui el sistema (a partir d'ara) s'engegarà en els nivells corresponents. Podem ara estar al nivell 5 i tenir el servei aturat perquè encara no l'hem engegat. Exemple:

```

[root@host ~]# runlevel
N 5
[root@host ~]# service sendmail status
sendmail està aturat

[root@host ~]# service sendmail start
S'està iniciant el servei sendmail [ FET ]

```

Els fitxers de monitorització dels serveis es troben usualment en el directori */var/log*. En el cas del servei *sendmail* es desen varis fitxers de registre anomenats *maillog-<data>*. Existeix també un directori anomenat */var/log/mail* amb la informació estadística de funcionament del servei.

Observar els registres de monitorització o logs del sistema:

```

# Fitxers de logs del servei sendmail
[root@host ~]# ll /var/log/mail*
-rw-r--r-- 1 root root 3545 24 gen 21:29 /var/log/maillog
-rw-r--r-- 1 root root 7269 25 des 19:41 /var/log/maillog-20111225
-rw-r--r-- 1 root root 12241 9 gen 18:28 /var/log/maillog-20120109
-rw-r--r-- 1 root root 8870 15 gen 13:42 /var/log/maillog-20120115
-rw-r--r-- 1 root root 11810 23 gen 17:13 /var/log/maillog-20120123

# Directori de logs de les dades de estadística
[root@host ~]# ll /var/log/mail
-rw-r--r-- 1 root root 728 24 gen 19:29 statistics

```

En el fitxer de monitorització del servei */var/log/maillog* es pot observar que queda enregistrat quan s'engega el servei, quan s'atura i també el registre dels emails que es transporten:

```

# extracte del fitxer /var/log/maillog:

Jun 13 18:23:40 portatil sendmail[4646]: m5DGNd0C004646: to=pere, ctld
dr=root (0/0), delay=00:00:01, xdelay=00:00:01, mailer=relay, pri=45837
, relay=[127.0.0.1] [127.0.0.1], dsn=2.0.0, stat=Sent (m5DGNdS5004647 M
essage accepted for delivery)

Jun 13 18:23:40 portatil sendmail[4648]: m5DGNdS5004647: to=<pere@tftp.
server.cat>, ctldaddr=<root@tftp.server.cat> (0/0), delay=00:00:00, xdel
ay=00:00:00, mailer=local, pri=46326, dsn=2.0.0, stat=Sent

Jun 13 18:57:11 portatil sendmail[4916]: alias database /etc/aliases re
built by root

Jun 13 18:57:11 portatil sendmail[4916]: /etc/aliases: 78 aliases, long
est 10 bytes, 787 bytes total

Jun 13 18:57:11 portatil sendmail[4921]: starting daemon (8.14.1): SMTP
+queueing@01:00:00

```

Un cop iniciat el servei es genera un bloqueig o *lock* amb el nom del servei per evitar que altres instàncies del servidor 'xoquin'. Podem observar el fitxer de *lock* fent:

```

[root@host ~]# ll /var/lock/subsys/sendmail
-rw-r--r-- 1 root root 0 13 jun 19:01 /var/lock/subsys/sendmail

```

Finalment observar el PID del procés del servidor:

```

[root@host ~]# ps ax | grep sendmail
5220 ?        Ss          0:00 sendmail: accepting connections
5228 ?        Ss          0:00 sendmail: Queue runner@01:00:00 for /var/spool/clientmqueue

```

El servei *sendmail* com la majoria de serveis stand-alone genera un fitxer amb el pid del procés en el directori */var/run* amb el nom del servei. Aquest fitxer conté únicament el PID del servei en *ascii*.


```
[root@host ~]# ll /var/run/sendmail.pid
-rw----- 1 root smmsp 33 13 jun 19:01 /var/run/sendmail.pid

[root@host ~]# cat /var/run/sendmail.pid
5220
/usr/sbin/sendmail -bd -qlh
```

Instal·lació en un sistema Debian

1.2.2. Comprovar el funcionament del servidor.

Comprovar que el servidor de correu està en funcionament és un procés ben senzill, n'hi ha prou de comprovar que el servei està engegat. Això no vol dir, en cap cas, que el servei estigui funcionant correctament. Potser el servidor està engegat però no està correctament configurat. De fet, la configuració és la part realment important de l'administració d'un servei.

A part de comprovar l'estat del servei (amb l'opció *status*), l'administrador pot assegurar-se que el dimoni del servei està en execució buscant el seu PID (*process identifier* o indicador de número de procés). Una altra activitat a fer és monitorar el registre d'activitats del servei (els *logs*). Tot el trànsit SMTP és trànsit de xarxa TCP/IP, per tant també es pot observar l'estat dels ports i analitzar el trànsit que s'hi produeix amb una eina de monitorització de xarxa.

Comprovació d'un servei

L'administrador pot verificar el funcionament del servidor de correu observant:

- l'estat del servei (on, off)
- el PID del servei (ha d'estar running).
- el fitxer de lock per evitar altres instàncies del servidor.
- el registre de *logs*.
- l'estat dels ports.
- monitorar el trànsit de xarxa amb una eina tipus *wireshark*, *netstat*, *iptraff* o *ss*.

Es pot observar l'estat del servei fàcilment amb l'ordre *service* en sistemes Fedora o amb */etc/init.d/sendmail* en sistemes Debian. Localitzar el PID del procés del servidor es pot fer usant ordres com *ps* o localitzar directament el fitxer del PID del servei. Finalment es pot observar que s'ha generat un fitxer de *lock* per evitar que es puguin iniciar altres instàncies del servidor.

```
# Mostrar l'estat actual del servei:
[user@host ~]# /etc/init.d/sendmail status
sendmail (pid 2697) s'està executant...
sm-client (pid 2695) s'està executant...

# Mostrar el PID del procés del servidor (buscar-lo per nom)
[user@host ~]# pgrep -l sendmail
2695 sendmail
2697 sendmail

# Filtrar els processos per nom
[user@host ~]# ps ax | grep sendmail
2695 ?      Ss      0:00 sendmail: Queue runner@01:00:00 for /var/spool/clientmqueue
2697 ?      Ss      0:00 sendmail: accepting connections

# Mostrar el fitxer del pid del servei
[user@host ~]# cat /var/run/sendmail.pid
2697
/usr/sbin/sendmail -bd -qlh

# Mostrar el fitxer de lock:
[user@host ~]# ll /var/lock/subsys/sendmail
-rw-r--r-- 1 root root 0 25 gen 16:26 /var/lock/subsys/sendmail
```

Tots els serveis del sistema normalment es monitoren anotant en fitxers de text un registre de totes les accions que realitzen, són els fitxers

coneguts com a fitxers de *log*. Es pot utilitzar un fitxer genèric pel sistema o bé un fitxer independent per a un servei determinat. El servidor de correu utilitza fitxers pròpis per monitorar els esdeveniments del servei, aquestes fitxers es troben en el directòri estàndard */var/log* i s'anomenen *maillog*. Es pot observar que existeixen diverses versions del fitxer corresponents a dates diferents.

rotació de logs:

El 'servei' logrotate és l'encarregat d'anar fent la rotació dels fitxers de log. En lloc de disposar d'un únic i gegantí fitxer amb tots els logs es van generant fitxers diferents per períodes de temps concrets (que es poden configurar). També es pot indicar quants fitxers antics cal conservar.

```
# Observar el fitxer de logs maillog i els històrics anteriors:
[root@host ~]# ls /var/log/maillog*
/var/log/maillog      /var/log/maillog-20120109  /var/log/maillog-20120123
/var/log/maillog-20111225 /var/log/maillog-20120115

# Observar el contingut del fitxer maillog
[root@host ~]# head -n5 /var/log/maillog
Jan 23 20:31:07 portatil sendmail[2463]: restarting /usr/sbin/sendmail due to signal
Jan 23 20:31:07 portatil sm-msp-queue[2461]: restarting /usr/sbin/sendmail due to signal
Jan 23 20:31:08 portatil sm-msp-queue[6205]: starting daemon (8.14.4): queueing@01:00:00
Jan 23 20:31:08 portatil sendmail[6207]: starting daemon (8.14.4): SMTP+queueing@01:00:00
Jan 23 21:49:21 portatil sendmail[1655]: starting daemon (8.14.4): SMTP+queueing@01:00:00

# Observar que el fitxer statistics és un fitxer amb dades binàries
[root@host ~]# file /var/log/mail/statistics
/var/log/mail/statistics: data
```

Sovint l'administrador vol comprovar que els ports que utilitza el protocol SMTP estan oberts. En GNU/Linux es pot fer una llista fàcilment dels serveis associats a cada port mitjançant el fitxer */etc/services*. Algunes utilitats com *nmap* permeten detectar els ports oberts. Es pot observar que el port usat pel servei SMTP és el port 25. Per exemple podem fer:

```
# Llista dels ports que inclouen alguna referència smtp:
[root@host ~]# cat /etc/services | grep smtp
smtp      25/tcp    mail
smtp      25/udp    mail
urd       465/tcp    smtps # URL Rendezvous Directory for SSM / SMTP over SSL (TLS)
rsmtpt    2390/tcp   # RSMTPT
rsmtpt    2390/udp   # RSMTPT
```

Monitorar el tràfic

Existeixen moltes eines per minitorar el tràfic de xarxa i l'estat dels ports, les més usuals són:

- *netstat* i *ss* com a eines de consola de text.
- *iptraf* com a eina de consola en mode finestres de text.
- *wireshark* és la eina omnipresent per analitzar de dalt a baix el tràfic de xarxa. És una eina imprescindible per observar què passa a la xarxa i ajudar a resoldre problemes de comunicació.

Wireshak

Tot i que sovint els alumnes interpreten wireshark com una eina 'atacant' perquè permet snifar el tràfic de xarxa i per exemple observar els passwords que viatgen en text pla, la veritat és que wireshark és una eina fantàstica d'ajuda a l'administrador per poder observar què passa a la xarxa i ajudar a detectar errors de configuració.

Utilitat netstat: Aquesta utilitat mostra el tràfic tcp, els ports i l'estat de la connexió:

```
[root@host ~]# netstat -tnvc
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.1.34:43223      212.170.21.168:25      ESTABLISHED

Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.1.34:43223      212.170.21.168:25      ESTABLISHED
```

Com es pot observar del llistat anterior, el client ha fet una connexió des del seu port dinàmic 42327 al port 25 del servidor. La connexió encara és activa com indica l'estat ESTABLISHED. Les opcions de "netstat -tnvc" serveixen per indicar que monitoritzi únicament connexions tcp, mostri el número de port i s'executi continuament per tal de monitoritzar el tràfic.

Utilitat ss: Podem llistar els ports amb l'ordre ss, com per exemple

```
[root@host ~]# ss
State      Recv-Q  Send-Q  Local Address:Port          Peer Address:Port
ESTAB      0        0       192.168.1.34:43223          212.170.21.168:smtp

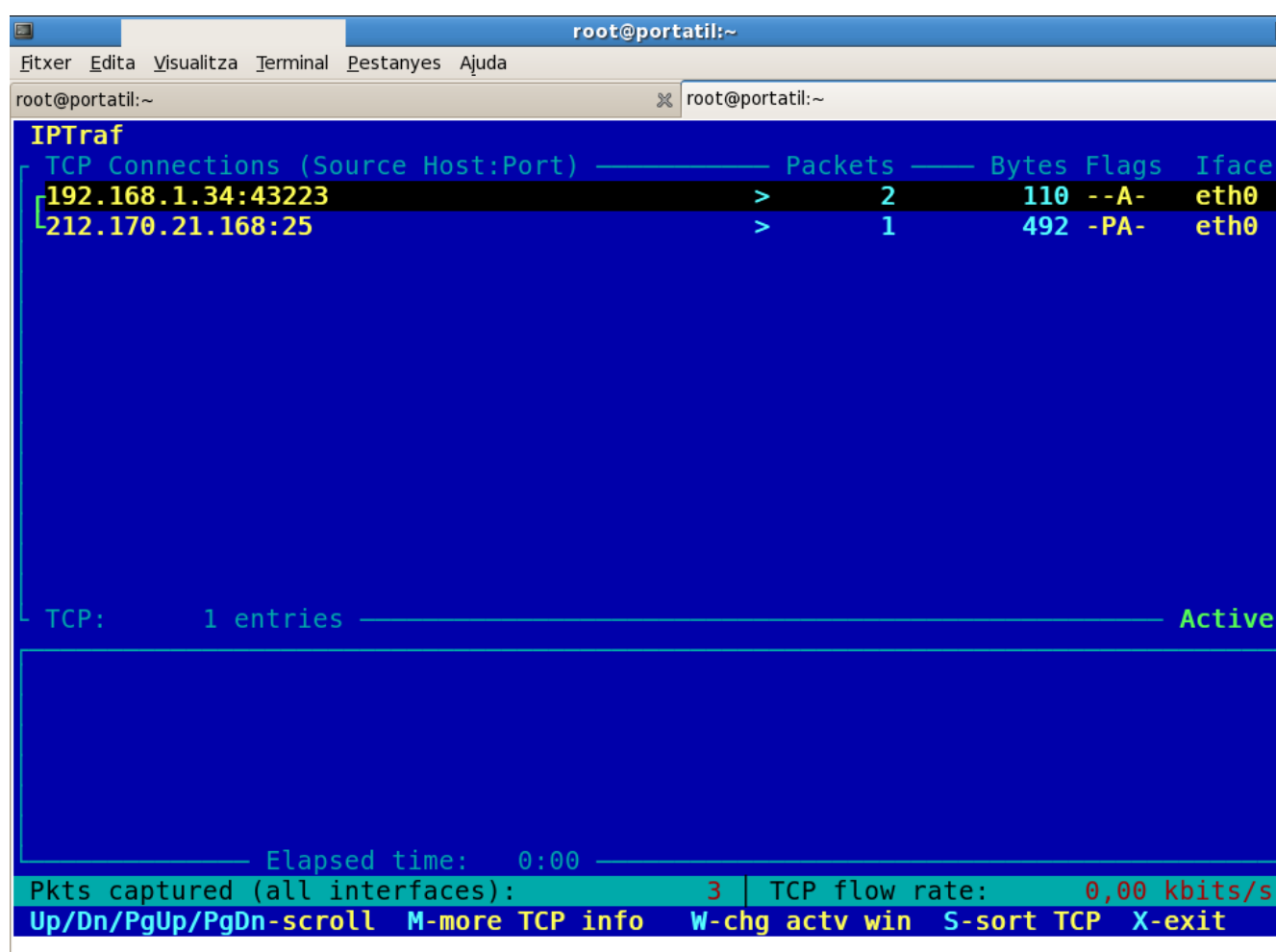
[root@host ~]# ss -nta
State      Recv-Q  Send-Q  Local Address:Port          Peer Address:Port
LISTEN     0        0       127.0.0.1:8000              *:
LISTEN     0        0       127.0.0.1:2208              *:
LISTEN     0        0       *:111                       *:
LISTEN     0        0       *:51125                     *:
LISTEN     0        0       127.0.0.1:2207              *:
ESTAB      0        0       192.168.1.34:43223          212.170.21.168:25
```

Fixeu-vos que les connexions al port 25 apareixen com a ESTAB ja que la connexió encara és oberta.

Utilitat iptraf: permet observar el tràfic TCP utilitzant una aplicació de text amb pantalles. La Figura 02 “Pantalla de monitorització de tràfic SMTP usant iptraf” mostra una connexió SMTP on es pot veure la connexió del client de sortida usant el port dinàmic 43223 al port 25 del servidor 212.170.21.168.

```
# Activar l'utilitat de consola iptraf
[root@host ~]# iptraf
```

Figura 2. Pantalla de monitorització de tràfic SMTP usant iptraf.



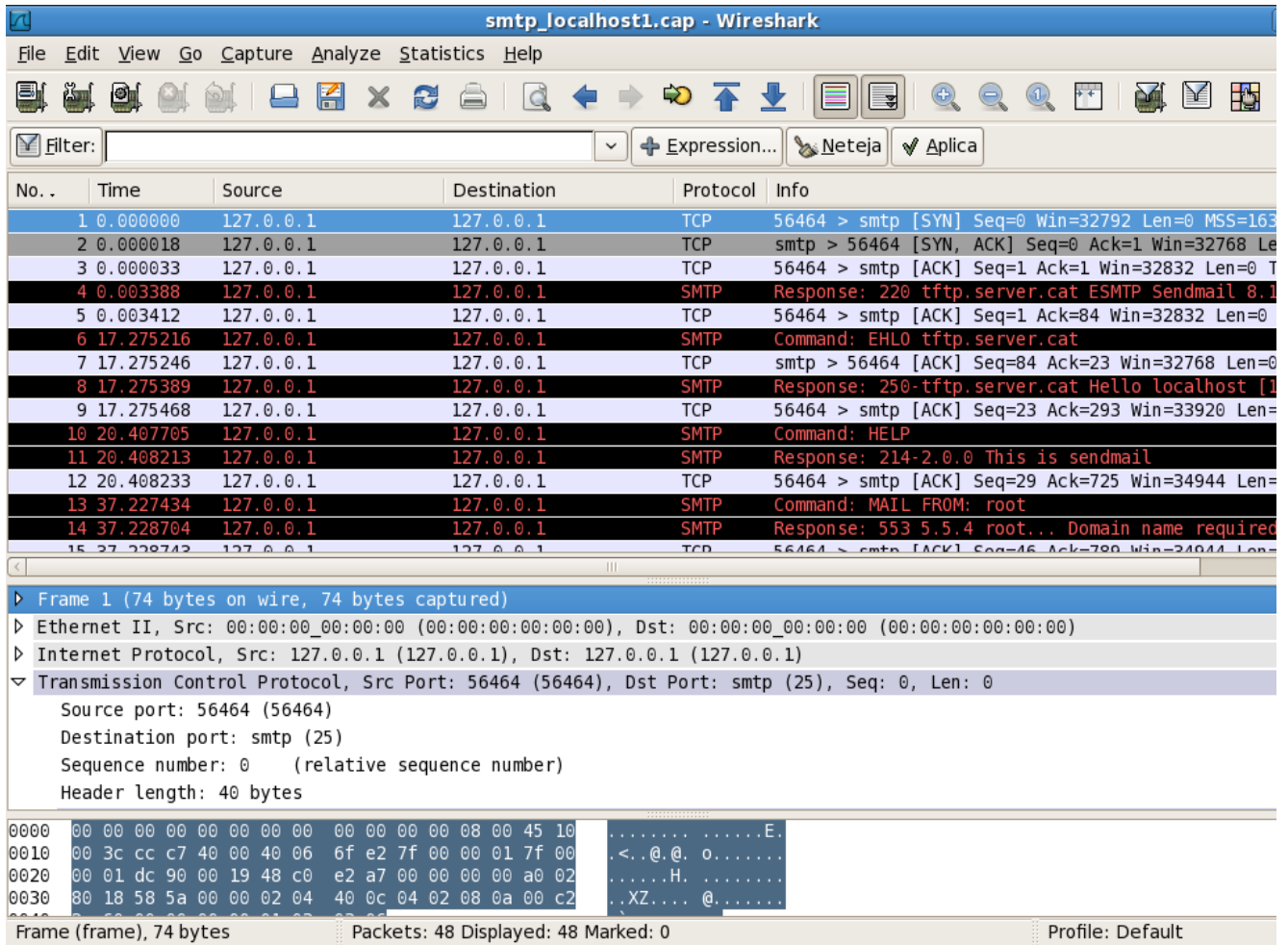
Utilitat wireshark: si s'utilitza Wireshark per monitoritzar el tràfic d'una connexió SMTP es podran observar els ports client (dinàmics) i ports del servidor (25), el tipus de tràfic TCP, la connexió TCP de tres vies, les peticions del client i les respostes del servidor. La Figura 3 “Pantalla de captura del tràfic SMTP utilitzant l'utilitat wireshark” mostra una captura de pantalla amb un diàleg SMTP. Podeu obtenir una captura del wireshark del fitxer del material complementari:

- Complementari_ASX_uf3_a1_smtp_dialeg.cap.

Figura 3. Pantalla de captura del tràfic SMTP utilitzant l'utilitat wireshark.

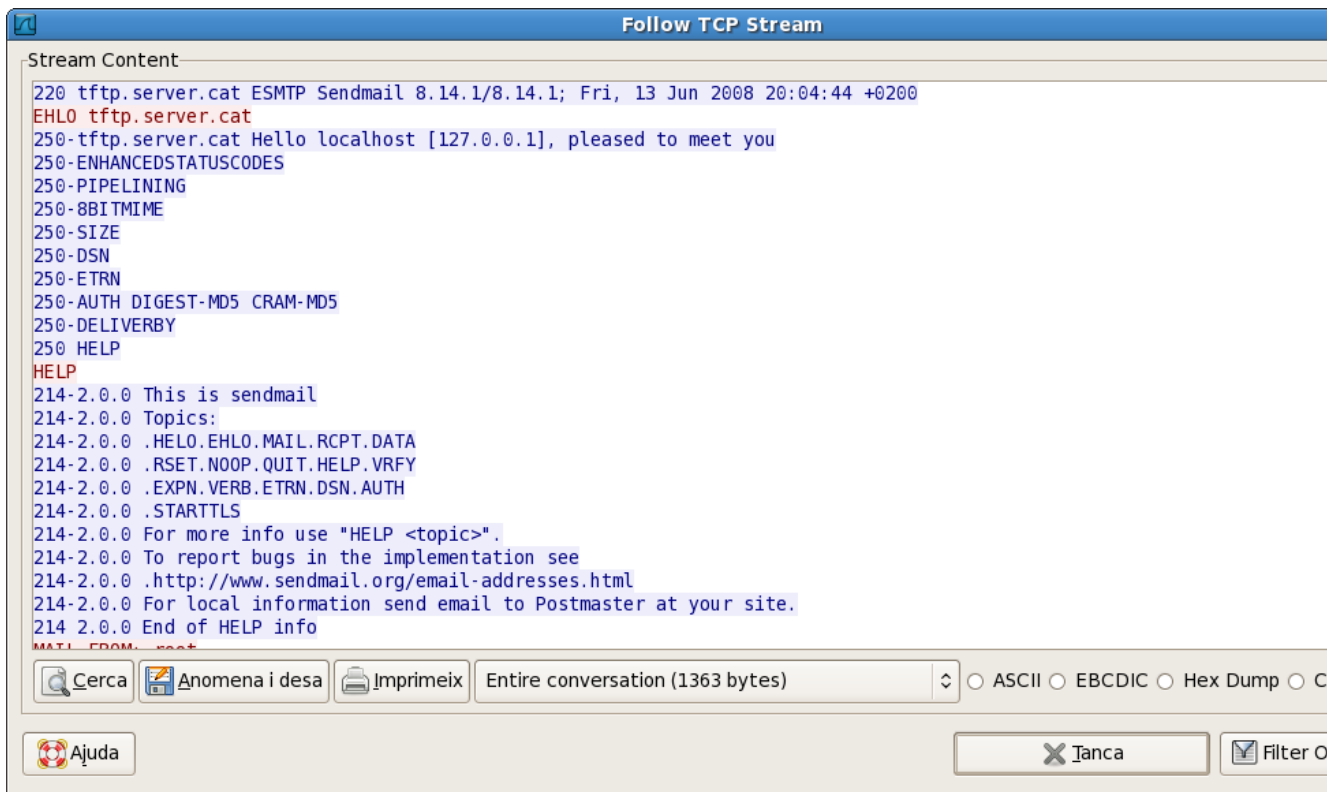
!!

En el material Complementari trobareu la captura d'un diàleg SMTP realitzada amb wireshark. Es pot importar la captura per tal d'estudiar-ne el tràfic.



Si es selecciona algun dels elements de la conversa TCP i s'activa l'opció contextual "follow TCP stream" es pot veure tot el diàleg TCP en text pla, és a dir, es pot veure tot allò que el client i el servidor de correu s'han estat dient, inclòs el contingut dels missatges. La Figura 4 "Pantalla de seguiment d'un diàleg SMTP en text pla, capturat amb wireshark" permet observar aquest diàleg.

Figura 4. Pantalla de seguiment d'un diàleg SMTP en text pla, capturat amb wireshark.



1.2.3.Sessió SMTP usant telnet

Si fem un breu repàs al funcionament del servei de correu recordarem que els missatges de correu s'envien de MTA a MTA usant el protocol SMTP. El servidor de correu (el programa *sendmail* en el cas exposat aquí) realitza tant la tasca de client com la de servidor. És a dir, el *sendmail* que vol enviar el missatge és qui contacta (fent la funció de client) amb el *sendmail* del destinatari (que fa la funció de servidor) per transferir-li el missatge. Tot aquest diàleg és en format TCP/IP i per tant pot ser simulat usant la utilitat *telnet*.

El servidor de correu **sendmail** realitza el paper de **MTA client SMTP** connectant a un altre host per transferir-li un correu. En aquest host destinatari s'executa un servidor de correu fque realitza el paper de **MTA servidor SMTP**.

Per tant, en instal·lar *sendmail* disposem tant de la funcionalitat client SMTP com de la funcionalitat de servidor SMTP.

En altres apartats d'aquest material s'ha vist com instal·lar i verificar el funcionament del servidor a nivell d'administració de sistemes (paquets, pid, logs, etc). Anem ara a veure com verificar el funcionament com a client i com a servidor smtp. Cal verificar les dues funcionalitats:

- **Client SMTP:** per verificar que el servei instal·lat funciona apropiadament com a client SMTP es pot realitzar un diàleg *telnet* amb qualsevol altre servidor de correu que existeixi a internet. Preferiblement un que sapiguem que ens deixa entrar i generar correu.
- **Servidor SMTP:** es pot verificar el funcionament del servidor simulant un diàleg SMTP usant *telnet* al pròpi host on hi ha instal·lat el servei (usualment *localhost*), i enviant un missatge a un usuari local.

Exemple 01: Client SMTP (sendmail)

En la següent sessió es pot observar una connexió al servidor de correu de l'escola del treball de barcelona:

```
[user@host ~]# telnet www.escoladeltreball.org 25
```

```

Trying 212.170.21.168...
Connected to www.escoladeltreball.org.
Escape character is '^]'.
220 escoladeltreball.org ESMTP Sendmail 8.13.8/8.13.8;
    Sat, 26 Apr 2008 19:56:05 +0200
EHLO escoladeltreball.org
250-escoladeltreball.org Hello 106.Red-81-39-13.dynamicIP.rima-tde.net
    [81.39.13.106], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 10000000
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250-DELIVERBY
250 HELP

HELP
214-2.0.0 This is sendmail version 8.13.8
214-2.0.0 Topics:
214-2.0.0      HELO    EHLO    MAIL    RCPT    DATA
214-2.0.0      RSET    NOOP    QUIT    HELP    VRFY
214-2.0.0      EXPN    VERB    ETRN    DSN     AUTH
214-2.0.0      STARTTLS
214-2.0.0 For more info use "HELP <topic>".
214-2.0.0 To report bugs in the implementation send email to
214-2.0.0      sendmail-bugs@sendmail.org.
214-2.0.0 For local information send email to Postmaster at your site.
214 2.0.0 End of HELP info

NOOP
250 2.0.0 OK

EXPN automocio@correu.escoladeltreball.org
502 5.7.0 Sorry, we do not allow this operation
EXPN baxillerat01@escoladeltreball.org
502 5.7.0 Sorry, we do not allow this operation

MAIL FROM: pepito@palotes.org
553 5.1.8 pepito@palotes.org... Domain of sender address
    pepito@palotes.org does not exist
MAIL FROM: user@xtec.cat
250 2.1.0 user@xtec.cat... Sender ok
RCPT TO: user@correu.escoladeltreball.org
250 2.1.5 user@correu.escoladeltreball.org... Recipient ok

RCPT TO: pepito@palotes.org
550 5.7.1 pepito@palotes.org... Relaying denied. Proper authentication required.
RCPT TO: user1@gmail.com
550 5.7.1 user1@gmail.com... Relaying denied. Proper authentication required.
RCPT TO: user1@escoladeltreball.org
250 2.1.5 user1@escoladeltreball.org... Recipient ok

DATA
354 Enter mail, end with "." on a line by itself
hola,
aquest és un missatge de prova per enviar un
email usant telnet al servidor smtp de l'escola.
s'envia una còpia a dos usuaris locals al servidor.
S'ha denegat fer relaying i enviar una còpia a
l'exterior
usuari
.
250 2.0.0 m3QH5B3012660 Message accepted for delivery

QUIT
221 2.0.0 escoladeltreball.org closing connection
Connection closed by foreign host.

```

Si seguim el desenvolupament de la sessió s'observarà que es realitzen les següents accions:

- Connectar al servidor de correu amb un telnet per simular el diàleg SMTP.
- El client envia un HELO per iniciar el diàleg. Es tracta d'un EHLO perquè el servidor implementa extensions del protocol.
- El client emet un HELP per saber quines comandes implementa el servidor.
- L'ordre NOOP que emet el client no fa res, però força el servidor a contestar.
- El client intenta obtenir la llista d'usuaris que formen part dels alias indicats, però el servidor no ho permet. Utilitza la comanda EXPN.
- El client s'identifica amb la comanda MAIL FROM, però com que no és un usuari amb un compte de correu local en el servidor, és rebutjat. Tot seguit s'identifica amb un usuari vàlid.
- El client ha d'indicar un a un els destinataris del missatge (pot usar alias i llistes de correu). Per a cada destinatari utilitza la comanda RCPT TO. Com que els dos destinataris primers no són destinataris locals (el primer és inventat i el segon és de gmail) el servidor els rebutja. Els rebutja perquè el servidor està configurat per no fer **"Relaying"**. Si es configura per acceptar el Relay, acceptaria els

destinatari i s'encarregaria de posar-se en contacte amb els servidors de correu dels dominis dels destinataris.

- Només un cop el servidor a acceptat almenys un destinatari es pot enviar el missatge. El client envia l'ordre DATA per indicar que enviarà a continuació el contingut del missatge. Tot seguit s'escriu tot el missatge (si es volen capçaleres també cal escriure-les) i s'acaba amb una línia que conté únicament un punt i un <crLf>.
- Finalment el client tanca la connexió amb l'ordre QUIT.

Si s'observa el diàleg detingudament es fa evident que el servidor extern amb el que s'ha contactat no permet emetre missatges procedents d'usuaris que no existèixin. Els missatges que s'envien a aquest servidor han d'anar destinats forçosament a un usuari que tingui bústia en aquest servidor. Per tant desestima els missatges destinats a usuaris d'altres dominis. Aquesta configuració evita que aquest servidor sigui una font de *spam*.

Un servidor configurat per fer **relaying** permet enviar missatges de tot tipus d'usuaris a altres dominis. Aquests tipus de servidors són la font principal de missatges de **spam**.

Una política més apropiada és verificar que l'emissor del missatge existeixi (així s'evita la suplantació) i verificar que el destinatari és un dels comptes de correu pròpis del servidor.

Exemple 02: Servidor SMTP (sendmail)

En la següent sessió es pot observar una connexió al servidor de correu local instal·lat anteriorment. S'utilitza un telnet al port 25 del localhost per contactar amb la pròpia màquina i provar així el funcionament del sendmail com a servidor SMTP.

```
[user@host ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 portatil ESMTP Sendmail 8.14.1/8.14.1; Sat, 26 Apr 2008 20:15:00 +0200

HELP
214-2.0.0 This is sendmail
214-2.0.0 Topics:
214-2.0.0      HELO    EHLO    MAIL    RCPT    DATA
214-2.0.0      RSET    NOOP    QUIT    HELP    VRFY
214-2.0.0      EXPN    VERB    ETRN    DSN     AUTH
214-2.0.0      STARTTLS
214-2.0.0 For more info use "HELP <topic>".
214-2.0.0 To report bugs in the implementation see
214-2.0.0      http://www.sendmail.org/email-addresses.html
214-2.0.0 For local information send email to Postmaster at your site.
214 2.0.0 End of HELP info

NOOP
250 2.0.0 OK

MAIL FROM: root@localhost.localdomain
250 2.1.0 root@localhost.localdomain... Sender ok
RCPT TO: superman@localhost.localdomain
250 2.1.5 superman@localhost.localdomain... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Subject: prova de correu local
Hola,
prova de missatge enviat a un usuari local
que és un alias (superman) de root .
root
.
250 2.0.0 m3QIF0pn003979 Message accepted for delivery

MAIL FROM: root@portatil
250 2.1.0 root@portatil... Sender ok
RCPT TO: user@xtec.cat
250 2.1.5 user@xtec.cat... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Subject: prova de usuari remot, fent relaying
hola,
prova de un email enviat usant un altre MTA que permet
relay. Aquest MTA es podria utilitzar com a spammer
root
.
250 2.0.0 m3QIF0po003979 Message accepted for delivery

QUIT
221 2.0.0 localhost.localdomain closing connection
Connection closed by foreign host.
```

En aquesta sessió SMTP es pot observar que s'envien dos missatges de correu un a un usuari local i un altre a un usuari extern. La seqüència és la següent:

- Connectar al servidor de correu amb un telnet per simular el diàleg SMTP.
- El client envia un HELO per iniciar el diàleg. Es tracta d'un EHLO perquè el servidor implementa extensions del protocol.
- El client emet un HELP per saber quines comandes implementa el servidor.
- L'ordre NOOP que emet el client no fa res, però força el servidor a contestar.
- El client s'identifica amb la comanda MAIL FROM amb un compte de correu local del servidor. Amb l'ordre RCPT TO indica un destinatari local que de fet és un alias de l'usuari root. Finalment escriu el contingut del missatge utilitzant l'ordre DATA.
- El següent email l'envia el mateix usuari però el destinatari és un usuari remot. Com que el servidor de correu local està configurat acceptant Relay el que farà és enviar el missatge al servidor de correu destí. El servidor de correu destí és el que gestiona el domini de correu del destinatari (xtec.cat) i que conté un usuari local amb el nom de compte indicat.
- Finalment el client tanca la connexió amb l'ordre QUIT.

El seguiment del diàleg SMTP anterior permet observar que la configuració per defecte del servidor accepta **RELAY** i no verifica l'existència de l'usuari emisor. De fet es podrien enviar missatges en nom d'altres usuaris (reals o imaginaris) destinats a usuaris d'altres dominis.

1.3.Crea i verifica l'accés als comptes d'usuari.

Un servidor de correu realitza la funcionalitat de client SMTP i de servidor SMTP. Com a client s'encarrega d'enviar els missatges que hi ha a la cua de missatges al servidor apropiat. És a dir, si un missatge va destinat a un usuari del domini *gmail.com* ha d'encarregar-se de fer arribar el missatge a algun dels servidors de correu d'aquest domini.

Com a servidor SMTP té la funció d'escoltar les peticions entrants que li fan els clients SMTP i atendre-les. Això significa 'rebre' els missatges i fer els passos necessaris per fer-los arribar a la bústia del client. Si el sistema de correu no utilitza un MDA pròpi (ho són programes com *procmail*, *spamassassin*, etc) serà el pròpi *sendmail* qui farà aquesta funció.

El servidor de correu quan actua com a **Servidor SMTP** pot també realitzar la funcionalitat de MDA (Mail Delivery Agent) i encarregar-se de deixar els missatges en la bústia de cada usuari local.

De fet si la configuració de correu actual que s'està utilitzant és la que ve per defecte en instal·lar *sendmail*, no hi ha cap MDA específic sinó que el pròpi *sendmail* fa aquesta funció. Això significa que també s'ha d'encarregar de crear les bústies de correu dels usuaris i de gestionar-les (a no ser que ja ho faci el pròpi sistema operatiu).

És missió del servidor de correu la creació de les **bústies d'usuari** i també la seva gestió si no hi ha altres agents tipus MDA en funcionament.

Un **compte** de correu no és més que disposar d'una bústia de correu en el sistema.

És evident que cada usuari que vol tenir un “compte de correu” ha de disposar d'una *bústia* pròpia on el servidor ha de poder desar-hi el correu destinat a l'usuari. L'usuari accedirà a la seva bústia per tal de poder consultar el seu correu. Des del punt de vista de la creació de bústies d'usuari podem fer la classificació següent:

- **Usuaris locals del sistema:** en sistemes GNU/Linux tots els usuaris del sistema disposen d'una bústia de correu local. Què cal fer perquè els usuaris d'un servidor tinguin correu? Res. Tots els usuaris locals d'un host tenen una bústia pròpia i tothom s'hi pot adreçar indicant **usuari@host**. Aquest mecanisme obliga a generar comptes d'usuaris locals en el sistema per tal de poder disposar dels comptes de correu.
- **Usuaris del servei:** hi ha servidors de correu que permeten crear 'comptes de correu' sense necessitat de crear els comptes d'usuari locals en el sistema. És a dir, es tracta d'usuaris que existeixen només per al servidor de correu però no per al sistema operatiu.

Els següents llistats de codi mostren els passos necessaris per crear dos usuaris locals en un sistema operatiu GNU/Linux, en un host anomenat *host*. El fet de crear els usuaris implica que es crearà la seva bústia de correu en format *mbox*. Aquest format posa tots els missatges en un únic fitxer de text amb el nom de l'usuari en el directori */var/spool/mail*.

```
# Crear els usuaris "pere" i "anna"
[root@host ~]# useradd -d /tmp/ pere
[root@host ~]# useradd -d /tmp anna
[root@host ~]# passwd pere
[root@host ~]# passwd anna

# Comprovar que el servei és actiu:
[root@host ~]# service sendmail status
sendmail (pid 5220) s'està executant...
sm-client (pid 5228) s'està executant...

# Comprovar que ni l'usuari pere ni anna tenen encara correu:
[pere@host ~]$ ll /var/spool/mail/pere
-rw-rw---- 1 pere mail 0 13 jun 19:16 /var/spool/mail/pere
[pere@host ~]$ mail
No mail for pere

[anna@host ~]$ ll /var/spool/mail/anna
-rw-rw---- 1 anna mail 0 13 jun 19:16 /var/spool/mail/anna
[anna@host ~]$ mail
No mail for anna
```

usuaris locals

si es vol que un usuari local no pugui iniciar una sessió d'usuari en el sistema sempre s'hi pot assignar com a shell */sbin/nologin*.

usuari@host

tot usuari de sistemes GNU/Linux disposa d'un compte local en la màquina on té el compte d'usuari. Així un usuari de nom *pere* en un host anomenat *pc-jocs* pot rebre correu a l'adreça *pere@pc-jocs*.

1.3.1. Verificar el compte de correu.

En aquest apartat es realitzaran varis exemples per mostrar la utilització (i per tant verificació) dels comptes de correu dels usuaris creats anteriorment. Els exemples han de permetre verificar els següents aspectes:

- Verificar que un usuari local és capaç d'enviar correu a altres bústies de correu locals (incloent la seva pròpia bústia).
- Verificar que un usuari local és capaç d'enviar correu a destinataris remots.
- Des d'un servidor de correu remot, per exemple un webmail com *gmail* enviar un correu a l'usuari local. Evidentment això només funcionarà si el compte de correu de l'usuari és en un servidor accessible des d'internet. Una prova més modesta, si disposem d'almenys dos ordinadors en una xarxa local, és enviar un correu d'un pc a un altre pc.

Exemple 01: enviar correu a comptes locals

Un usuari d'un sistema GNU/Linux pot enviar correus a altres usuaris utilitzant la utilitat *mail*, que existeix des de temps immemorials.

mail és una utilitat en mode text dels sistemes Unix i GNU/Linux que permet enviar missatges de correu a destinataris locals i remots.

També permet als usuaris accedir a la seva bústia local i consultar el seu correu. De fet permet gestionar la bústia local llistant, eliminant i filtrant missatges.

Els següents fragments de codi mostren que l'usuari pere envia un missatge a l'usuària anna (es fa des d'una sessió de l'usuari pere). En una altra sessió l'usuària anna observa que ha rebut un missatge i consulta la seva bústia de missatges. Selecciona el primer (i únic) missatge i comprova que és el mateix que li ha enviat en pere. *nota: observeu que el nom de domini del host és *tftp.server.cat*.

```
# Enviar missatge de pere a anna:
[pere@host ~]$ mail -s "missatge de pere a anna" anna
hola
aquest es un email de prova de pere a anna
bye!
.
Cc:

# Comprovar que anna l'ha rebut:
[anna@host ~]$ll /var/spool/mail/anna
-rw-rw---- 1 anna mail 641 13 jun 19:21 /var/spool/mail/anna
You have new mail in /var/spool/mail/anna

[anna@host ~]$mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/anna": 1 message 1 new
>N 1 pere@tftp.server.cat Fri Jun 13 19:21 18/641 "missatge de "
& 1
Message 1:
From pere@tftp.server.cat Fri Jun 13 19:21:59 2008
Date: Fri, 13 Jun 2008 19:21:58 +0200
From: pere@tftp.server.cat
To: anna@tftp.server.cat
Subject: missatge de pere a anna

hola
aquest es un email de prova de pere a anna
bye!
```

Exemple 02: enviar correu a comptes remots

La mateixa utilitat *mail* vista en l'exemple anterior permet enviar missatges a usuaris remots. Una prova fàcil de fer és enviar un missatge a un compte de correu d'algun webmail com per exemple *Gmail* (evidentment a un compte de correu vostre per poder-ho verificar!). En l'exemple següent l'usuària anna envia un missatge de correu a un compte de *gmail* d'un usuari anomenat "ecanet1profe". En la Figura 5 "Pantalla del webmail de correu Gmail" es pot observar que l'usuari ha rebut el missatge enviat per l'anna.

```
[anna@host ~]$mail -s "missatge a l'exterior de anna" ecanet1profe@gmail.com
aquest missatge s'ha de lliurar a l'exterior,
a un compte de gmail
adeu!
.
Cc:
```

Figura 5. Pantalla del webmail de correu Gmail.

Exemple 03: Rebre localment missatges d'usuaris remots

L'últim cas a verificar és comprovar que es reben a les bústies locals dels usuaris pere i anna missatges procedents de comptes d'usuaris remots. El mecanisme més 'bonic' de fer seria enviar un missatge des d'un webmail com per exemple *Gmail* a l'usuari pere. Perquè això sigui possible cal que el host que conté la seva bústia sigui accessible des d'internet. Això implica:

- que el host pertany a un domini públic d'internet. És a dir, que és adreçable via DNS des d'internet.
- que el host és el servidor de correu declarat d'aquest domini (o un d'ells). És a dir, que el servei DNS té etiquetat aquest host com una entrada *MX* Mail exchanger.

Evidentment aquest és un exemple complex de fer per els alumnes. Un exemple més senzill és enviar correu d'un pc a un altre en una xarxa local. Segurament a l'aula de classe o a casa es disposa almenys de dos ordinadors. La prova consisteix en enviar un correu des d'un pc, per exemple "pc01" al pc que té els comptes d'en pere i l'anna, anomenat "host".

Per referenciar un compte de correu en una màquina remota en una xarxa local usar l'anotació **usuari@maquina** on màquina pot ser el nom de *host* (cal resolució) o la seva *adreça IP*.

El següent fragment de codi mostra com l'usuari root del "pc01" envia un missatge a l'usuari pere que està en un equip anomenat "host". Un cop rep el missatge en pere accedeix a la seva bústia per llegir-lo.

```
# Enviar missatge des de root del pc01 pere@host:
[root@pc01 ~]$ mail -s "missatge de root per a pere" pere@host
hola
aquest es un email de prova de root del pc01 a pere de l'ordinador "host"
bye!
.
Cc:
# Comprovar que en pere l'ha rebut:
```

Per poder referenciar un ordinador pelseu nom cal que hi hagi un servei de resolució DNS o de manera més simple que tingui una entrada local al fitxer de *hosts*. Un altre mecanisme és usar l'adreça IP.

```
[pere@host ~]$ll /var/spool/mail/pere
-rw-rw---- 1 pere mail 641 13 jun 19:21 /var/spool/mail/pere
You have new mail in /var/spool/mail/pere

[pere@host ~]$mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/pere": 1 message 1 new
>N 1 root@pc01 Fri Jun 13 19:21 18/641 "missatge de root per a pere"
```

1.3.2.Format dels missatges i les bústies de correu.

Els missatges que rep un usuari es depositen en la seva bústia de correu. Per tot usuari 'real' existeix la seva corresponent bústia de correu, però no per tot email, ja que es poden crear àlies i llistes de distribució que en realitat apunten a altres comptes de correu. Com són i on es troben aquestes bústies de correu?. Existeixen diferents formats de bústies, és dir diferents maneres d'emmagatzemar els missatges d'un compte de correu:

- **mbox:** aquest és el format clàssic usat en els sistemes GNU/Linux consistent en un sol fitxer de text pla o es van desant acumuladament (concatenats) els missatges de correu que rep l'usuari. Típicament es troba un fitxer amb el nom de l'usuari al directori *spool* del correu.
- **maildir:** aquest nou format desa els missatges en un directori per a cada usuari, desant cada missatge de correu en un fitxer pròpi i permetent generar una estructura de directòris en arbre per organitzar els continguts de la bústia.
- **les bústies dels webmails:** usualment el sistema de correu amb el que està més habituat l'usuari és amb el *webmail*, correu web com del de *Gmail* o *Yahoo*. L'estructura interna que utilitzen aquests proveïdors gratuïts de correu és desconeguda per l'usuari. De fet es desconeix si és en format de un sol fitxer de text, un arbre de directòris, en una base de dades, etc. Ara bé, des del punt de vista del client que utilitza la web per gestionar el seu correu tots aquests serveis acostumen a proporcionar una estructura similar: el *inbox* o safata d'entrada, el *sent* o carpeta de missatges enviats, el *spam* o carpeta on van aparar els missatges spam o *correu brossa*, *draft* amb els missatges a mig el-laborar, etc.

Tot seguit s'analitzarà el format i les bústies de correu. Es conegut que els missatges de l'especificació original es componen únicament de text pla i per poder incorporar continguts binaris s'utilitza l'especificació MIME, que requereix una codificació del missatge per tal de generar text. S'analitzarà els següents aspectes:

- Format dels missatges al mbox.
- Missatges amb adjunts (MIME).
- Missatges de resposta (MIME).
- Missatges de text no ascii pur (codificacions).
- Format de missatges MIME des d'un webmail.
- Codificació de continguts MIME.

Format dels missatges al mbox

Anem a analitzar com es desen els missatges dins d'una bústia de tipus mbox. Els missatges es desen en text pla (codificats si cal) concatenats, separats els uns dels altres per una línia en blanc. Cada missatge comença amb la informació de l'*envelop* FROM, a continuació hi ha totes les capçaleres, una línia en blanc per separar les capçaleres del cos del missatge i finalment tot allò que diu el cos del missatge.

El següent fragment de codi s'assegura d'esborrar la bústia de missatges de l'usuari pere i omplir-la de nou enviant-li un missatge:

```
[root@host ~]# rm /var/spool/mail/pere
rm: voleu eliminar el fitxer ordinari «/var/spool/mail/pere»? y

[root@portatil ~]# mail -s "mail simple" pere
missatge enviat de l'usuari root a l'usuari pere
el missatge es desarà al mailbox de l'usuari pere a /var/spool/mail/pere
el missatge no conte cap atachment
adeu!
.
Cc:
```

Tot seguit es llista el contingut del fitxer de text pla que realitza la funció de bústia de correu:

```
[root@portatil ~]# cat /var/spool/mail/pere
From root@tftp.server.cat  Fri Jun 13 17:11:04 2008
Return-Path: <root@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
    by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DFB3bK003856
    for <pere@tftp.server.cat>; Fri, 13 Jun 2008 17:11:04 +0200
Received: (from root@localhost)
    by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DFB332003855
    for pere; Fri, 13 Jun 2008 17:11:03 +0200
Date: Fri, 13 Jun 2008 17:11:03 +0200
From: root <root@tftp.server.cat>
Message-Id: <200806131511.m5DFB332003855@tftp.server.cat>
To: pere@tftp.server.cat
Subject: mail simple

missatge enviat de l'usuari root a l'usuari pere
el missatge es desarà al mailbox de l'usuari pere a /var/spool/mail/pere
el missatge no conte cap attachment
adeu!
```

Com es pot observar en el mailbox hi ha únicament un missatge que conté les següents capçaleres:

- *FROM* és la part del sobre que indica el destinatari. Fixeu-vos que la capçalera no porta “:” per tant, és la part del sobre o envelop. És a dir, no és una capçalera del missatge sinó un indicador de la bústia de que en aquest punt comença un missatge nou.
- La capçalera *Return-Path* forma part del cos del missatge i indica l'adreça del receptor.
- Les capçaleres *From:* i *To:* són capçaleres del cos del missatge i indiquen l'emissor i el receptor (fixeu-vos amb els “:”).
- Els camps *Received* indiquen el recorregut que ha fet el missatge a través dels MTA (el que es llista primer és el més recent o proper al destinatari, el que es llista últim és el més proper a l'emissor). Com es pot observar el missatge el rep sendmail de l'usuari root i s'envia al servidor tftp.server.cat on hi ha l'usuari pere.
- La capçalera *Date:*, *Message-ID:* i *Subject:* indiquen la data en que el MTA rep el missatge per enviar-lo, l'identificador únic del missatge i el tema del missatge.

Missatges amb adjunts (MIME)

El següent llistat mostra una bústia de correu que conté un missatge amb dos fitxers adjunts, un pdf i una imatge jpeg.

```
[pere@host ~]$ cat /var/spool/mail/pere
From root@tftp.server.cat  Fri Jun 13 17:26:31 2008
Return-Path: <root@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
    by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DFQTH7003922
    for <pere@tftp.server.cat>; Fri, 13 Jun 2008 17:26:30 +0200
Received: (from root@localhost)
    by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DFQSIq003918
    for pere@tftp.server.cat; Fri, 13 Jun 2008 17:26:28 +0200
Date: Fri, 13 Jun 2008 17:26:27 +0200
From: root <root@portatil.local.lan>
To: pere@tftp.server.cat
Subject: missatge amb atachment
Message-ID: <20080613152627.GA3909@portatil.local.lan>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="zYM0uCDKw75PZbzx"
Content-Disposition: inline
User-Agent: Mutt/1.5.17 (2007-11-01)
Status: 0

--zYM0uCDKw75PZbzx
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline

missatge de root a l'usuari pere
conte adjunt un pdf i jpeg
adeu!

--zYM0uCDKw75PZbzx
Content-Type: application/pdf
Content-Disposition: attachment; filename="informatica_AX_ud2.pdf"
Content-Transfer-Encoding: base64
... output suprimir (correspón al contingut del pdf codificat en base64) ...

--zYM0uCDKw75PZbzx
Content-Type: image/jpeg
Content-Disposition: attachment; filename="cd15_11_puerto-madrin.jpg"
Content-Transfer-Encoding: base64
... output suprimir (correspón al contingut del jpeg codificat en base64) ...

--zYM0uCDKw75PZbzx--
```

mbox:

en una bústia *mbox* els missatges es separen entre ells per una línia en blanc (excepte el primer) i una línia inicial amb el “from” de l'envelop. Un “From ” sense els dos punts.

Com es pot observar el missatge s'identifica com a contingut de tipus MIME:

- La capçalera MIME Type: indica que es tracta d'un missatge MIME.
- La capçalera *Content-Type: multipart/mixed; boundary = "zYM0uCDKw75PZbzx"* indica que el contingut és multipart. El camp boundary és l'identificador que permetrà diferenciar cada part del contingut multipart.
- Es pot observar que la primera part s'identifica per l'etiqueta: *"zYM0uCDKw75PZbzx "* que separa cada part. Cada separador comença amb *"-"*. La primera part és el text del propi email i per tant el contingut és text pla tal i com indica la capçalera *"Content-Type: text/plain; charset=us-ascii"*.
- La següent part (indicada altre cop pel separador) conté el fitxer adjunt que és de tipus PDF tal i com indica la capçalera *"Content-Type: application/pdf"*. Com que el contingut no es transportable pel protocol SMTP cal una codificació. S'utilitza la codificació base64 tal i com indica la capçalera *"Content-Transfer-Encoding: base64"*.
- La última part es troba a continuació del separador de parts (*-zYM0uCDKw75PZbzx*) i conté el contingut jpeg com indica la capçalera *"Content-Type: image/jpeg"*. El contingut també es codifica en base64.
- S'indica el final de les diferents parts quan es troba un separador que comença i acaba amb *"-"* (de fet quan acaba). Es pot observar que el email acaba amb *"-zYM0uCDKw75PZbzx- "*.

Missatges de resposta (MIME)

Molts dels emails que enviem són resposta a d'altres emails que s'han rebut. Els missatges de resposta són de tipus MIME i poden ser de diversos formats diferents. El més simple i usual és incloure la resposta com a text, que apareix al final destacada d'un color diferent o identada amb el caràcter *">"*. Les respostes incorporen un camp que permet identificar el missatge al que responen pel seu ID.

És una pèssima pràctica de treball adjuntar els missatges en les respostes. No fa més que fer créixer la mida del missatge com una bola de neu.

El codi següent mostra un missatge de resposta que ha enviat l'usuari pere a l'usuari root tot adjuntant el missatge original.

```
From pere@tftp.server.cat Fri Jun 13 17:47:33 2008
Return-Path: <pere@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
    by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DFLWC0004001
    for <root@tftp.server.cat>; Fri, 13 Jun 2008 17:47:32 +0200
Received: (from pere@localhost)
    by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DFLUAU003997
    for root@tftp.server.cat; Fri, 13 Jun 2008 17:47:30 +0200
Date: Fri, 13 Jun 2008 17:47:30 +0200
From: pere@portatil.local.lan
To: root <root@tftp.server.cat>
Subject: Re: mail simple
Message-ID: <20080613154730.GA3987@portatil.local.lan>
References: <200806131511.m5DFB332003855@tftp.server.cat>
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
In-Reply-To: <200806131511.m5DFB332003855@tftp.server.cat>
User-Agent: Mutt/1.5.17 (2007-11-01)

Aquest es l'email de resposta de l'usuari pere
a l'usuari root. Inclou el missatge original.
adeu.

On Fri, Jun 13, 2008 at 05:11:03PM +0200, root wrote:
> missatge enviat de l'usuari root a l'usuari pere
> el missatge es desara al mailbox de l'usuari pere a /var/spool/mail/p
ere
> el missatge no conte cap attachment
> adeu!
```

Ens podem fixar que en ser un missatge de resposta a un email anterior apareix:

- Una capçalera *References*: que identifica el missatge del que és resposta.
- Conté els camps pertinents que descriuen que es tracta d'un missatge MIME. En aquest cas com que tant l'original com la resposta són text pur el missatge no és multipart sinó simplement text ascii.

Missatges de text no ascii pur (codificacions)

Els missatges de correu no poden contenir caràcters que no formen part del conjunt de caràcters del codi ascii de 7 bits (els 128 primers

caràcters) ni caràcters de control (els 31 primers caràcters). Un dels mecanismes de codificació de missatges és indicar el missatge com a *quoted-printable*, és a dir, es codifiquen els caràcters no imprimibles i es mantenen igual tots els caràcters 'clàssics'.

El següent codi mostra un missatge que conté caràcters de text que no són ascii pur i com s'ha codificat.

```
[root@host ~]# mail -s "email amb caracters no ascii" pere
hola
aquest email l'envia l'usuari root a l'usuari pere.
Conté caràcters especials i d'accentuació, com per
exemple accents i símbols estranys com @ i #.
Com ho tracta el MUA?
.
Cc:

[root@host ~]# less /var/spool/mail/pere
From root@tftp.server.cat  Fri Jun 13 17:55:55 2008
Return-Path: <root@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
    by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DFtsjL004031
    for <pere@tftp.server.cat>; Fri, 13 Jun 2008 17:55:55 +0200
Received: (from root@localhost)
    by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DFts7l004030
    for pere; Fri, 13 Jun 2008 17:55:54 +0200
Date: Fri, 13 Jun 2008 17:55:54 +0200
From: root <root@tftp.server.cat>
Message-Id: <200806131555.m5DFts7l004030@tftp.server.cat>
To: pere@tftp.server.cat
Subject: email amb caracters no ascii

hola
aquest email l'envia l'usuari root a l'usuari pere.
Conté caràcters especials i d'accentuació, com per
exemple accents i símbols estranys com @ i #.
Com ho tracta el MUA?
```

Format de missatges MIME des d'un webmail

El format dels missatges es pot visualitzar també des dels webmail. Així si volem observar quina és l'estructura de capçaleres i cos d'un missatge rebut per exemple al *Gmail* també es pot fer. No és una opció massa popular entre els usuaris però si es busca amb atenció s'observarà que entre mig de les típiques opcions de "contestar, reenviar, filtrar, esborrar..." també hi sol haver l'opció de mostrar el *contingut original* del missatge. La Figura 6 "Opcions de gmail per mostrar el contingut original d'un missatge" mostra les opcions de *Gmail* que permeten mostrar el contingut original del missatge. El contingut original es pot observar en la Figura 7 "Pantalla de gmail que mostra el contingut original d'un missatge." es poden observar les capçaleres i el contingut del missatge MIME rebut.

Figura 6. Opcions de gmail per mostrar el contingut original d'un missatge.

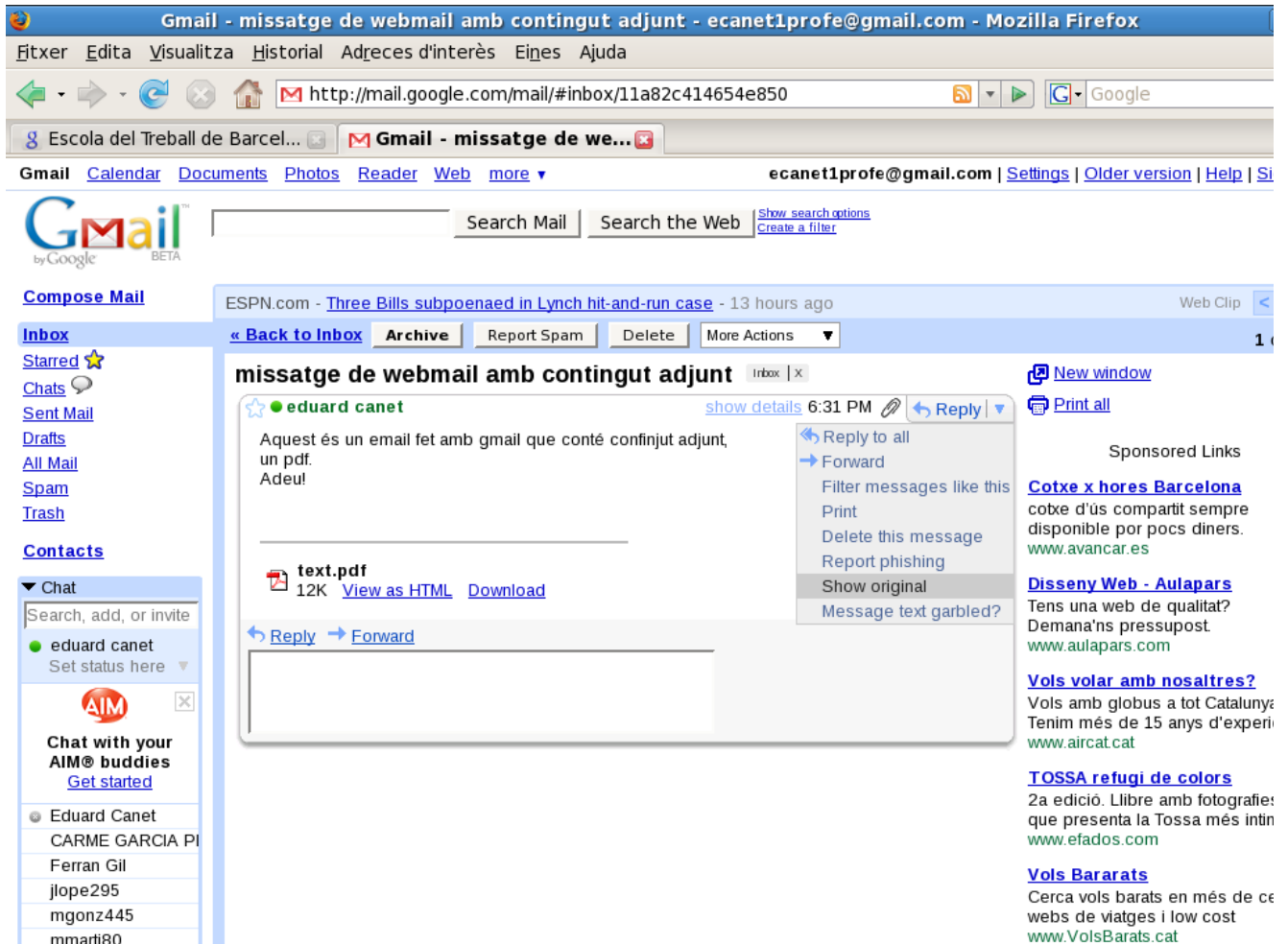
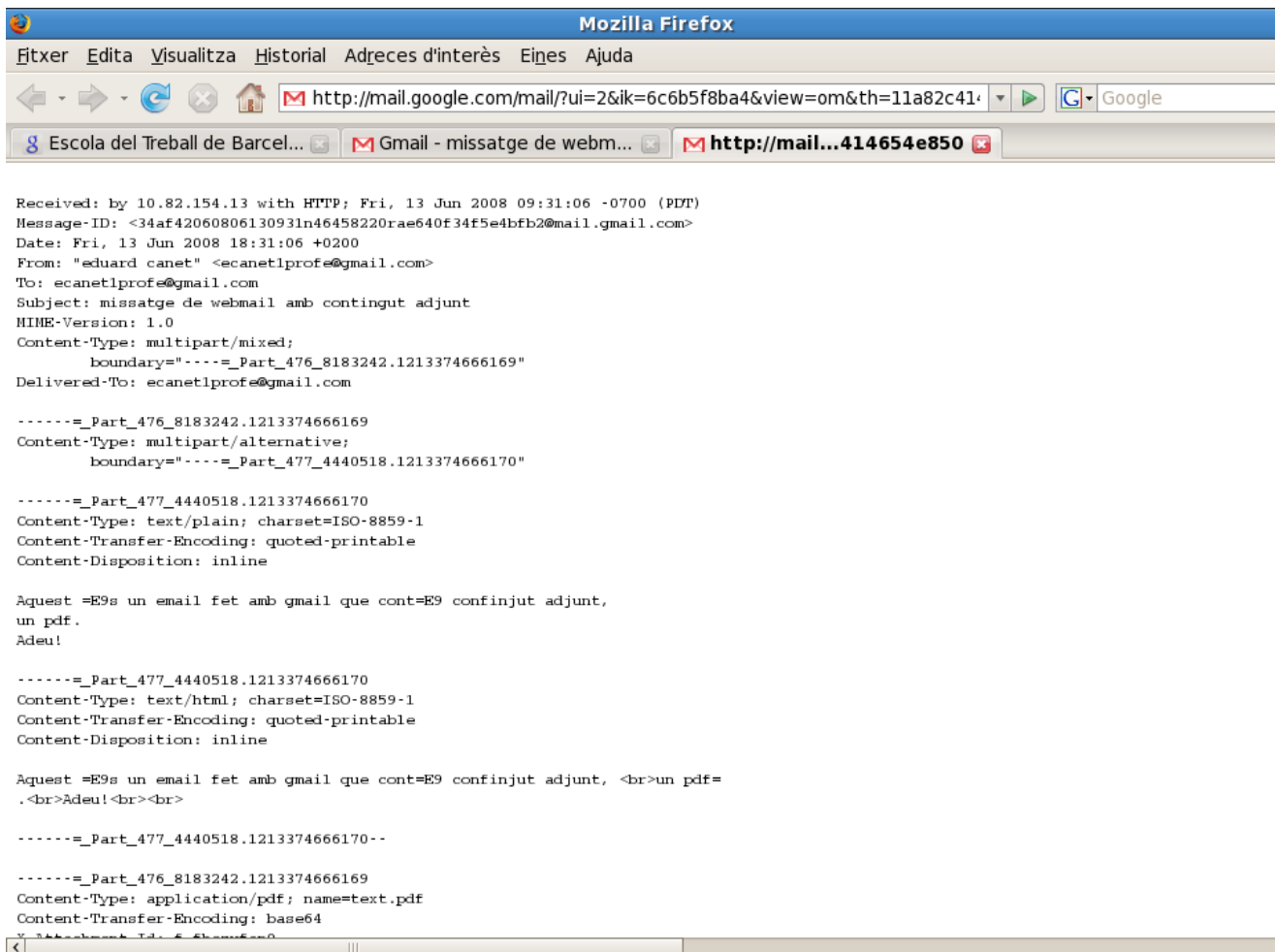


Figura 7. Pantalla de gmail que mostra el contingut original d'un missatge.



Codificació de continguts MIME

La codificació que s'utilitza generalment en el sistema de correu per enviar contingut que no és text pur és codificació *base64*. Es pot fer una prova de com funciona aquesta codificació per enviar un petit fitxer pdf usant mail. El següent codi mostra com transformar un fitxer pdf a un fitxer de text codificat a *base64*.

```
# Convertir el text pdf al format base64:
[pere@host ~]# base64 text.pdf > text.pdf.b64

[pere@host ~]# ll
-rw-r--r-- 1 pere pere 11685 13 jun 18:06 text.pdf
-rw-r--r-- 1 pere pere 15785 13 jun 18:09 text.pdf.b64

# El fitxer text.pdf.b64 es pot transferir com a text pla.
[pere@host ~]# mail -s "enviar un pdf codificat a base64" pere < text.pdf.b64

# El receptor pot extreure el contingut i desar el fitxer codificat:
[pere@host ~]$ mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/pere": 5 messages 4 unread
U 1 pere@tftp.server.cat Fri Jun 13 18:11 221/16399 "enviar un pd"

& s 1 nou.pdf.b64
"nou.pdf.b64" [New file]

# S'ha desat el missatge en un fitxer
[pere@host ~]$ ll nou.pdf.b64
-rw-rw-r-- 1 pere pere 16400 13 jun 18:14 nou.pdf.b64

# Decodificar el missatge per obtenir el pdf
[pere@host ~]$ base64 -d nou.pdf.b64 > nou.pdf
```

1.4. Estableix i aplica mètodes per impedir usos indeguts del servidor de correu electrònic.

El principal problema del correu electrònic avui en dia és el correu brossa o *spam*, és a dir, el correu no desitjat. Des del punt de vista client convé saber filtrar el correu per detectar l'*spam* i informar-ne al

servidor (avui en dia el webmail). Des del punt de vista del servidor cal saber filtrar el correu brossa i cal saber establir mecanismes per no participar en la difusió del correu spam.

Actualment la majoria de serveis de correu de tipus webmail incorporen les dues prestacions següents:

- **Filtrat automàtic d'spam:** el pròpi *gmail* per exemple filtra els correus i intenta detectar quins són d'spam i els posa directament en una carpeta del mateix nom. El servei de *gmail* aplica regles complexes de filtrat per detectar quins correus són en la seva opinió spam. Els usuaris poden ajudar informant de quins missatges són spam i això ajuda a millorar l'eina de filtrat.
- **Xequeig automàtic de virus:** aquesta és una altra de les excel·lents característiques que es proporcionen, aplicar un antivirus als continguts que s'adjunten als fitxers. D'aquesta manera s'evita la propagació indiscriminada de continguts maliciosos.

Aquestes característiques es poden afegir a la majoria de servidors web que hi ha al mercat. El més usual és que el servidor no proporcioni directament aquesta funcionalitat sinó afegir-li un mòdul o cridar un software a part que s'encarregui de portar a terme aquestes tasques per al servidor. En sistemes GNU/Linux algunes de les eines populars per processar els missatges són:

- **procmail:** és una utilitat que permet establir regles de processament de missatges (les famoses 'receptes de cuina'). Per exemple donada una determinada condició (missatge secret de James Bond) enviar-ne una còpia a tots els mitjans de comunicació (còpia a múltiples destinataris), o per exemple si el missatge diu la paraula "factura hipoteca" destruir-lo immediatament!.
- **spamassin:** és una eina que permet aplicar regles i informació provinent d'una base de dades de coneixement d'spam per tal de detectar missatges de correu no desitjat.

L'altra objectiu importantíssim que ha de tenir en ment l'administrador d'un servidor de correu és evitar que el servidor formi part del sistema (il·lícit) de difusió de missatges d'spam. El correu brossa normalment es genera des d'ordinadors d'usuaris 'innocents' que no saben que el seu equip s'està utilitzant (governat remotament per altres persones) per difondre spam. És per això que cal plantejar-se els següents mecanismes de control:

- **validar l'emissor:** una primera mesura de seguretat és validar que l'usuari emisor existeixi. En lloc d'acceptar els missatges sigui el que sigui l'emisor (com s'ha fet en diversos exemples d'aquest material), el servidor ha de verificar que el compte de correu de l'emisor existeixi. D'aquesta manera s'evita que es puguin generar missatges en nom d'usuaris fictícies.

El problema de l'emisor del missatge

Un dels problemes del mecanisme de correu és que quan un MTA client (host-A) transfereix els missatges a un servidor MTA (host-B) li va 'dictant' missatge a missatge d'aquells que van destinats a host-B. Per cada missatge informa del emisor, el destinatari(s) i el cos del missatge. El *FROM* indica qui és l'emisor del missatge i es pot escriure qualsevol cosa. Validar que existeixi no evita però que es pugui usar com a from el nom d'un 'altre' usuari vàlid.

Segurament heu rebut alguna vegada missatges d'spam que representa que els heu enviat vosaltres mateixos o els vostres amics, tot i no haver-ho fet, oi?

- **validar que el receptor és local:** una mesura més de seguretat *importantíssima* és que el servidor de correu únicament accepti missatges destinats a usuaris locals del servidor. D'aquesta manera s'evita que el servidor propagui missatges destinats a altres dominis. Un servidor que accepti l'encàrrec de lliurar missatges a altres dominis és un servidor que pot ser usat fàcilment per a generar spam.

Un servidor de correu a internet no ha d'estar configurat en mode **Relay**, és a dir, no ha d'acceptar fer lliurament de correu a altres dominis. Un servidor configurat per defecte fent *Relay* accepta l'encarrec de difondre els correus que li arriben a altres servidors sense validar-ne la procedència i el destinatari, evidentment aquesta és la causa principal de l'spam.

El següent fragment d'una conversa SMTP usant telnet permet observar que s'intenta enviar un missatge en nom de l'usuari "pepito@palotes.org" i el servidor el denega perquè verifica l'inexistència d'aquest compte d'usuari. També s'intenta enviar el correu a un usuari inexistent (el propi

Filtrat de missatges

La majoria d'usuaris desconeixen que el seu servei webmail proporciona opcions de filtrat personalitzades, és a dir, la possibilitat d'establir regles que permetin dirigir els missatges a una carpeta o una altra de la bústia.

MDA

Els dos agents que es presenten aquí: procmail i spamassin realitzen tasques de MDA, és a dir, agafen el missatge del MTA, el processen i el deixen a la bústia de correu de l'usuari.

“pepito”) i a un usuari de gmail. En tots dos casos el servidor contesta informant que no accepta el *Relaying*, és a dir, no vol fer d'encarregat d'enviar correus a usuaris que no són del seus.

```
[user@host ~]# telnet www.edt.org 25
Trying 212.170.21.168...
Connected to www.edt.org.
Escape character is '^]'.
220 edt.org ESMTP Sendmail 8.13.8/8.13.8;
  Sat, 26 Apr 2008 19:56:05 +0200
EHL0 edt.org
250-edt.org Hello 106.Red-81-39-13.dynamicIP.rima-tde.net

MAIL FROM: pepito@palotes.org
553 5.1.8 pepito@palotes.org... Domain of sender address
  pepito@palotes.org does not exist

RCPT TO: pepito@palotes.org
550 5.7.1 pepito@palotes.org... Relaying denied. Proper authentication required.

RCPT TO: user1@gmail.com
550 5.7.1 user1@gmail.com... Relaying denied. Proper authentication required.
```

El següent fragment de comunicació mostra com un servidor configurat en mode *relay* accepta qualsevol usuari com a emissor del missatge i qualsevol destinatari.

```
[user@host ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 portatil ESMTP Sendmail 8.14.1/8.14.1; Sat, 26 Apr 2008 20:15:00 +0200

MAIL FROM: root@localhost.localdomain
250 2.1.0 root@localhost.localdomain... Sender ok

RCPT TO: superman@localhost.localdomain
250 2.1.5 superman@localhost.localdomain... Recipient ok

RCPT TO: pimpampums@pataplim.ximpum.cat
250 2.1.5 pimpampums@pataplim.ximpum.cat... Recipient ok
```

1.5.Instal·la serveis per permetre la recollida remota del correu existent en les bústies d'usuari.

La forma en que els usuaris accedeixen al seu correu ha anat evolucionant al mateix pas que ho han anat fent les tecnologies. Originalment s'utilitzava simplement SMTP (*sendmail* per exemple) i els usuaris havien d'accedir al servidor realitzant una sessió d'usuari per poder consultar el correu amb eines com *mail*. És a dir, l'usuari havia d'anar on físicament hi havia la màquina servidor i iniciar-hi una sessió o bé connectar-s'hi via *telnet*. En qualsevol cas iniciava una sessió en el servidor per poder consultar i generar correu.

Amb la popularització d'internet els usuaris volen poder-se descarregar des de casa els missatges de correu i també poder-ne enviar des de casa. El propòsit POP d'accés remot a les bústies de correu proporciona aquest servei. Típicament els usuaris es connectaven per mòdem, es connectaven al servidor de correu i es descarregaven tot el correu de cop a casa. Aprofitaven també per enviar aquells missatges que fes falta. En aquest model la gestió dels missatges es fa a casa, el servidor simplement els acumula per permetre'n la descàrrega. La tecnologia d'accés a internet via mòdem implicava pagar per les trucades, per tant l'usuari tenia l'interès de baixar tot el correu, penjar la trucada (per no pagar) i examinar tranquil·lament els missatges a casa sense la connexió d'internet.

Amb l'aparició de les tarifes planes els usuaris ja no s'han de preocupar “de fer una connexió ràpida al servidor”. Poden estar connectats permanentment amb el servidor i mirar tot el correu mantenint la connexió de xarxa. El protocol IMAP d'accés a bústies remotes permet l'accés dels clients a les seves bústies realitzant tota la gestió de la bústia (carpetes, etiquetat, filtrat, etc) en el pròpi servidor. Això resol un dels típics problemes del POP que és que en baixar-se un usuari els missatges en màquines diferents quedava el correu mal repartit en diferents llocs.

Sempre que es configura un client de correu cal indicar:

- el servidor de **correu entrant**, que correspon a un servidor POP o IMAP d'on es descarreguen els missatges del usuari.

- el servidor de **correu sortint**, que correspon al servidor SMTP a qui cal lliurar el correu que genera l'usuari per tal de que sigui enviat al destinatari.

En els temps actuals la majoria de clients de correu utilitzen els serveis prestats per servidors webmail com *Gmail*, *yahoo*, etc. Ara els clients es connecten via web al servidor i accedeixen utilitzant una interfície web a la seva bústia. Tota la gestió dels missatges, carpetes, etiquetat, filtrat, etc, es realitza a través de l'interfície web usant un navegador.

1.5.1.El servei POP.

En el model de transport de correu SMTP s'exigeix que el receptor disposi de connexió permanent a Internet. Està pensat per a correu entre organitzacions connectades a Internet i que disposen d'un servidor de correu que conté les bústies dels usuaris locals de l'organització. Això obliga l'usuari a treballar localment en el servidor per accedir a les seves bústies. Però, en popularitzar-se Internet, sorgeix el problema dels usuaris que hi accedeixen per ISP i que no tenen connexió permanent (per exemple, amb mòdem). Com poden disposar del servei de correu?

S'ideja un mecanisme per a l'accés remot als comptes de correu, de manera que l'usuari es connecta quan vol (no permanent), accedeix a la bústia de correu per recuperar els missatges i finalitza la connexió. POP3 i IMAP són protocols que permeten l'accés remot de clients a les bústies de correu.

POP3 és la versió actual del protocol POP. Usarem tots dos noms indistintament.

POP3 (post office protocol o protocol d'accés simple a les bústies de correu) és un protocol de capa d'aplicació de la pila de protocols TCP/IP (port 110) definit en l'RFC 1939. Permet a un client de correu (MUA) obtenir remotament el correu dipositat en la bústia de correu de l'usuari en un servidor de correu POP3.

POP3, similar al correu postal

El POP3 és un mecanisme similar al correu postal. El carter deixa les cartes a la nostra bústia i les recollim quan ens sembla.

Normalment, l'usuari utilitza una aplicació client de POP3 (per exemple, Thunderbird) i baixa el correu del servidor POP. Els missatges que es baixen es desen en la màquina de l'usuari (localment) i s'esborren del servidor (es pot configurar si s'esborren o no). Finalment es tanca la connexió.

Funcionament del POP3

Fragmentació del correu

El funcionament generalitzat del protocol POP3 és que el client fa una connexió TCP/IP al port 110 del servidor, es baixa el correu i tanca la connexió. En aquest procés, client i servidor passen per tres estats (autorització, transacció i actualització) i s'intercanvien ordres i respostes seguint el model de diàleg de Telnet:

Si l'usuari baixa correu POP des de màquines diferents, el correu li queda fragmentat, desat localment en cada màquina el que s'hi ha baixat.

- **Ordres.** Les ordres POP3 són ordres de text de quatre caràcters seguides d'espais i els arguments que requereixin. Finalitzen amb un <crLf>.
- **Respostes.** Les respostes POP3 són una cadena de caràcters que comença per **+OK** o **-ERR** més una descripció. Les respostes afirmatives comencen per +OK, i les d'error per -ERR.

Tot seguit es mostra una llista de les ordres utilitzables en el protocol POP3 agrupades segons l'estat:

1) Autorització

- **USER nomUsuari.** El client s'identifica davant del servidor POP indicant el nom d'usuari, que ha de correspondre a una bústia de correu del servidor POP.
- **PASS password.** El client s'ha d'autenticar indicant un nom d'usuari i una contrasenya vàlids. L'ordre **PASS** permet indicar aquesta contrasenya en text net.

- **APOP nomUsuari password-md5.** Per proporcionar més seguretat en el procés d'autorització, l'usuari pot fer servir l'ordre APOP que té com a arguments el nom d'usuari i la contrasenya encriptada usant una funció resum o *hash* com per exemple "MD5".

2) Transacció

- **STAT.** Amb aquesta ordre l'usuari demana l'estat de la seva bústia. El servidor retorna el nombre de missatges que conté i el total de bytes que ocupa.
- **LIST [msg].** Llista els missatges o un missatge concret. No llista el contingut sinó que llista per a cada missatge el seu número de missatge i el nombre de bytes que ocupa.
- **RETR msg.** Baixa un missatge concret del servidor al client. Els missatges es poden indicar pel número de missatge.
- **DELE msg.** Marca el missatge indicat per ser esborrat. No s'esborra immediatament, només es marca, l'esborrament es produeix en l'estat final d'actualització. En els MUA que actuen de clients POP és típic permetre configurar si es deixen els missatges en el servidor o s'eliminen. Usualment s'eliminen perquè només hi quedin els nous.
- **NOOP.** Aquesta ordre no fa res, però força el servidor a emetre una resposta positiva. Serveix per comprovar que la connexió encara és oberta.
- **RSET.** Si es realitza aquesta ordre abans de passar a l'estat d'actualització, RSET desmarca tots els missatges que estaven marcats per esborrar.
- **TOP msg nLin.** En lloc de baixar tot un missatge sencer com fa l'ordre RETR, l'ordre TOP permet baixar les línies inicials d'un missatge. Baixa les capçaleres i les *nLin* línies inicials. Aquesta ordre és útil per baixar només les capçaleres (remittents, assumptes, etc.) i per filtrar els missatges a baixar i marcar-los per esborrar ja directament sense baixar-los.
- **UIDL [msg].** Els missatges s'identifiquen dins de la línia pel seu número d'ordre (com fa l'ordre LIST), però el número d'ordre d'un missatge pot variar entre connexions si s'esborren els precedents. Per tal de poder identificar de manera única un missatge independentment de la posició que ocupa, es pot usar el UID. El UID és únic per a cada missatge en una bústia. L'ordre UIDL llista tots els uids i el seu número d'ordre o només el d'un missatge en concret.
- **QUIT.** L'ordre quit indica al servidor que el client vol finalitzar la connexió. El servidor passa de l'estat de transacció al d'actualització.

Baixar missatges del servidor POP3

Alguns MUA bàsics baixen tots els missatges del servidor de cop. Si en el servidor es deixen els missatges ja llegits, es tornen a baixar cada cop.

3) Actualització

En aquest estat no hi ha ordres. El servidor elimina els missatges marcats per esborrar, emet una resposta positiva al client, i tots dos tanquen la connexió.

L'exemple següent correspon a un diàleg mitjançant Telnet entre client i servidor usant el protocol POP, on es poden veure les ordres i respostes del protocol.

```
[root@portatil ~]# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
+OK POP3 localhost 2007a.104 server ready
USER pere
+OK User name accepted, password please
PASS pere
+OK Mailbox open, 1 messages

STAT
+OK 1 480
NOOP
+OK No-op to you too!
LIST
+OK Mailbox scan listing follows
1 480
.
RETR 1
+OK 480 octets
Return-Path: <root@localhost.localdomain>
Received: from tftp.server.cat (localhost [127.0.0.1])
  by tftp.server.cat (8.14.1/8.14.1) with SMTP id m5DI4ig8005681
  for pere@tftp.server.cat; Fri, 13 Jun 2008 20:06:12 +0200
Date: Fri, 13 Jun 2008 20:04:44 +0200
From: root <root@localhost.localdomain>
```

```

Message-Id: <200806131806.m5DI4ig8005681@tftp.server.cat>
Status:
.
aquest es un email qualsevol
el text s'escrui fins acabar
amb una línia que només conté un punt
.
QUIT
+OK Sayonara
Connection closed by foreign host.

```

Hi ha diverses implementacions de servidors POP i cada una implementa un conjunt d'ordres i extensions pròpies. L'especificació POP3 requereix que s'implementin almenys les ordres següents: STAT, LIST, RETR, DELE, NOOP, REST.

El model POP3

El POP3 és un protocol que permet l'accés remot a les bústies de correu dels usuaris, com també ho és l'IMAP. Ni el POP3 ni l'IMAP transporten el correu, aquesta funció la fa el protocol SMTP. Podeu veure el model funcional del protocol POP en la Figura 1 "Model funcional del protocol SMTP" i també en la Figura 12 "Model funcional del procoal IMAP", en què un usuari amb el seu MUA baixa el correu del servidor mitjançant POP o IMAP.

Com la majoria de serveis de xarxa a Internet, el protocol POP3 s'estructura seguint l'esquema client/servidor. Aquests són els agents que intervenen en una comunicació POP3:

- **MUA (mail user agent o agent d'usuari de correu).** L'usuari interactua amb un agent d'usuari per accedir al correu mitjançant POP3. Són agents d'usuari programes com Thunderbird, Getmail, Fetchmail, MS Outlook Express, Eudora, Gmail, etc. Com es pot veure en la llista, les aplicacions MUA poden ser de text, gràfiques i fins i tot pel web. Aquestes aplicacions incorporen el programari necessari per actuar de clients POP3.
- **Client POP3.** La part pròpiament encarregada de comunicar amb el servidor POP3 per obtenir els missatges de la bústia de correu de l'usuari és el client POP3. Client i servidor POP3 parlen un llenguatge comú que és el protocol POP3.
- **Servidor POP3.** Per poder implementar l'accés remot al correu cal disposar d'un servidor POP3 en funcionament. Aquest servidor POP3 conté les bústies dels usuaris o hi accedeix. Els missatges es reben mitjançant SMTP, i és un MTA o un MDA qui els diposita en la bústia. El servidor POP3 atén les peticions dels clients POP3 per baixar el correu.

Un concepte que ajuda a diferenciar el funcionament de POP3 i IMAP és que en l'esquema de POP3 es considera que l'emmagatzematge del correu es realitza en l'usuari. El servidor acumula els missatges nous i aquests es baixen tots al host de l'usuari i s'eliminen del servidor. Per tant, és responsabilitat de l'usuari com els gestiona. Si bé és cert que els missatges es poden desar en el servidor (sense esborrar), no hi ha eines per gestionar-los, tots estan en una mateixa carpeta o **folder**. El servidor POP3 ofereix poques funcionalitats: baixar missatges, baixar les capçaleres i esborrar els missatges.

Una sessió POP3 passa per tres estats clarament diferenciats:

1) Autorització . Un cop feta la connexió TCP/IP pel port 110 entre el client i el servidor POP3, s'entra en l'estat d'autorització. Cal que el client s'identifiqui davant del servidor POP3 indicant el compte d'usuari i la contrasenya.

2) Transacció . Un cop el client ha estat autoritzat pel servidor, s'entra en l'estat de transacció. En aquest estat el client demana accions (ordres) al servidor i aquest les atén. És a dir, en aquest estat el client es baixa el correu, marca missatges per esborrar, demana les capçaleres dels missatges, en fa una llista per ordre, etc. El client finalitza l'estat de transacció utilitzant l'ordre quit.

3) Actualització . El servidor entra en l'estat d'actualització en rebre l'ordre quit del client. Elimina els missatges marcats per esborrar (fins ara no s'havien eliminat) i envia un ok al receptor. Ara tots dos ja poden finalitzar la comunicació.

Accés POP3 al correu web

Molts correus web o *webmails* permeten baixar correu d'altres web usant el protocol POP3 o IMAP. Per exemple, des del Gmail es poden baixar missatges de Yahoo.

1.5.2. Instal·lació del servei POP.

Tot seguit es descriurà el procés per instal·lar el servei POP en un entorn GNU/Linux. Un cop feta la instal·lació cal observar què s'ha instal·lat, quins programes executables, on són els fitxers de configuració, els de monitoratge, etc.

El procés és mimètic al descrit per instal·lar el servidor SMTP. Per treballar amb un servidor de correu POP i IMAP s'instal·larà el software *uw-imap* que proporciona totes dues funcionalitats. Es tracta del servidor elaborat per la Universitat de Washington.

La principal diferència entre aquest servidor *uw-imap* i els instal·lats anteriorment és que es tracta s'un servei de xarxa configurat per executar-se dins del super servei de xarxa *xinetd*.

Instal·lació en un sistema Fedora

Els següents fragments de codi mostren tot el procés per identificar, instal·lar i examinar els paquets del servidor POP i IMAP *uw-imap* usant un GNU/Linux Fedora.

Llistar els paquets que contenen la cadena *uw-imap* o *uw* i instal·lar el paquet corresponent al servidor. Si volem comprovar si el sistema ja té instal·lats aquests paquets podem consultar quins paquets *uw-imap* hi ha actualment instal·lats fent:

```
[root@host ~]# yum search uw-imap
# Buscar els paquets que contenen sendmail
# Instal·lar el paquet servidor
# Observar quins paquets instal·lats hi ha amb la cadena uw-imap
[root@host ~]# rpm -qa | grep sendmail
uw-imap-2007a1-3.fc7
```

Obtenir informació del paquet del servei *uw-imap*:

```
[root@host ~]# rpm -qi sendmail
Name      : uw-imap                      Relocations: (not relocatable)
Version   : 2007a1                     Vendor: Fedora Project
Release   : 3.fc7                     Build Date: dj 15 mai 2008 13:49:10 CEST
Install Date: dv 13 jun 2008 20:40:30 CEST Build Host: xenbuilder4.fedora.phx.redhat.com
Group     : System Environment/Daemons Source RPM: uw-imap-2007a1-3.fc7.src.rpm
Size      : 135643                     License: ASL 2.0
Signature : DSA/SHA1, ds 17 mai 2008 01:24:23 CEST, Key ID b44269d04f2a6fd2
Packager  : Fedora Project
URL       : http://www.washington.edu/imap/
Summary   : UW Server daemons for IMAP and POP network mail protocols
Description:
The uw-imap package provides UW server daemons for both the IMAP (Internet
Message Access Protocol) and POP (Post Office Protocol) mail access
protocols. The POP protocol uses a "post office" machine to collect
mail for users and allows users to download their mail to their local
machine for reading. The IMAP protocol allows a user to read mail on a
remote machine without downloading it to their local machine.
```

Observar els components del paquet

Fer la llista dels components del paquet *uw-imap*:

```
[root@host ~]# rpm -ql uw-imap
/etc/pam.d/imap
/etc/pam.d/pop
/etc/pki/tls/certs/imapd.pem
/etc/pki/tls/certs/ipop3d.pem
/etc/xinetd.d/imap
/etc/xinetd.d/imapd
/etc/xinetd.d/ipop2
/etc/xinetd.d/ipop3
/etc/xinetd.d/pop3s
/usr/sbin/imapd
/usr/sbin/ipop2d
/usr/sbin/ipop3d
/usr/share/doc/uw-imap-2007a1
/usr/share/doc/uw-imap-2007a1/SSLBUILD
/usr/share/man/man8/imapd.8uw.gz
/usr/share/man/man8/ipopd.8uw.gz
```

En funció del directori on s'ubiquen els fitxers podem intuir si són executables, de configuració o de documentació. També podem mirar de filtrar la sortida en cada cas.

Fitxers de configuració:

```
[root@host ~]# rpm -qlc uw-imap
/etc/pam.d/imap
/etc/pam.d/pop
/etc/pki/tls/certs/imapd.pem
/etc/pki/tls/certs/ipop3d.pem
/etc/xinetd.d/imap
/etc/xinetd.d/imapd
/etc/xinetd.d/imap2
/etc/xinetd.d/ipop2
/etc/xinetd.d/ipop3
/etc/xinetd.d/pop3s
```

Fitxers de documentació:

```
[root@host ~]# rpm -qld uw-imap
/usr/share/doc/uw-imap-2007a1
/usr/share/doc/uw-imap-2007a1/SSLBUILD
/usr/share/man/man8/imapd.8uw.gz
/usr/share/man/man8/ipopd.8uw.gz
```

Podem mirar de filtrar quins són els executables tenint en compte que usualment estaran en un directori de nom *bin* o *sbin*. En aquest cas hi ha un únic executable corresponent al dimoni del servei.

```
[root@host ~]# rpm -ql uw-imap | grep "bin"
/usr/sbin/imapd
/usr/sbin/ipop2d
/usr/sbin/ipop3d
```

En resum:

- Els fitxers de documentació es troben generalment a: */usr/share/doc* i a */usr/share/man*.
- Executable: */usr/sbin/ipop3d*. S'identifica perquè pertany a un directori d'executables.
- Configuració: */etc/xinetd.d/ipo3* i */etc/xinetd.d/pop3s*. El fitxer de configuració del servei és un fitxer de configuració dins del superservei de xarxa xinetd.
- Tipus de servei: El dimoni *uw-imap* actua dins del superservei de xarxa xinetd. El xinetd s'executa i escolta les peticions POP entrants i en rebre-les enrega el servidor uw-imap.

Ubicació de fitxers

En GNU/Linux els fitxers executables per l'administrador es troben usualment a */sbin* i a */usr/sbin*. Els executables d'usuari normalment són a */bin* i */usr/bin*.

Activar/desactivar el servei i establir els nivells d'arrencada

Un cop s'ha instal·lat al sistema un servei de xarxa, cal posar-lo en funcionament. Primer caldrà determinar quin tipus de servei és: si autònom o integrat dins del superservei de xarxa. Un cop fet això, cal saber si ja està en funcionament o no. De fet cal saber engegar-lo, aturar-lo i reinicialitzar-lo. Finalment cal establir quin estat ha de tenir el servei per defecte cada cop que s'engegui el servidor.

• El servei

Primerament cal saber si el servidor instal·lat funciona *"stand-alone"* o dins del superdimoni de xarxa *"xinetd"* o *"initd"*. Si existeixen fitxers de configuració dins del directori */etc/xinetd.d/<nom-servei>* es tracta d'un servei dins del *xinetd*. Si existeixen fitxers de configuració dins del directori */etc/rc.d/init.d/<nom-servei>* es tracta d'un servei *"stand-alone"*.

Observem ara el contingut del paquet per intentar saber si els fitxers de configuració ens permeten saber de quin tipus de servei es tracta:

```
[root@host ~]# rpm -ql uw-imap | grep xinetd
/etc/xinetd.d/imap
/etc/xinetd.d/imapd
/etc/xinetd.d/ipop2
/etc/xinetd.d/ipop3
/etc/xinetd.d/pop3s
```

Com podem observar es tracta d'un servei *stand-alone*. També es pot consultar el tipus de servei amb l'ordre *chkconfig* i observar si surt la llista d'un tipus o de l'altre:

```
[root@hoat ~]# chkconfig --list | grep pop
ipop2:          apagat
ipop3:          apagat
pop3s:          apagat
```

• Estat del servei

Els serveis de xarxa que funcionen dins del *xinetd* es configuren "dins" d'un directori anomenat *xinetd.d*. Per cada servei existeix un fitxer de configuració:


```
[root@hosr ~]# ll /etc/xinetd.d/
... output suprimit ...
-rw-r--r-- 1 root root 370 26 set 2005 imap
-rw-r--r-- 1 root root 365 26 set 2005 imaps
-rw-r--r-- 1 root root 453 26 set 2005 ipop2
-rw-r--r-- 1 root root 359 26 set 2005 ipop3
-rw-r--r-- 1 root root 335 26 set 2005 pop3s
```

El format d'aquests fitxers de configuració és similar:

```
[root@host ~]# cat /etc/xinetd.d/ipop3
# default: off
# description: The POP3 service allows remote users to access their mail \
#               using an POP3 client such as Netscape Communicator, mutt, \
#               or fetchmail.
service pop3
{
    socket_type          = stream
    wait                 = no
    user                  = root
    server                = /usr/sbin/ipop3d
    log_on_success        += HOST DURATION
    log_on_failure        += HOST
    disable               = yes
}
```

Per activar el servei calen dos passos:

- Activar el servei del servidor POP dins del fitxer de configuració del xinetd. Usualment això significa editar la línia del fitxer `/etc/xinetd.d` `</servei>` i modificar l'entrada que diu “`disable=yes`” per “`disable=no`”.
- Engegar el servei xinetd.

Activar el servei POP dins del xinetd:

```
[root@host ~]# vi /etc/xinetd.d/ipop3
# default: off
# description: The POP3 service allows remote users to access their mail \
#               using an POP3 client such as Netscape Communicator, mutt, \
#               or fetchmail.
service pop3
{
    socket_type          = stream
    wait                 = no
    user                  = root
    server                = /usr/sbin/ipop3d
    log_on_success        += HOST DURATION
    log_on_failure        += HOST
    disable               = no
}
```

Per activar el servei cal executar l'script de gestió del servei:

```
[root@host ~]# service xinetd start
[root@host ~]# /etc/rc.d/init.d/xinetd start
S'està iniciant el xinetd:
```

• Establir els nivells d'execució

Activar per defecte el servei *xinetd* per als nivells 3, 4 i 5 d'execució del sistema.

```
# Mostrar l'estatus del servei xinetd per a cada runlevel
[root@host ~]# chkconfig --list xinetd
xinetd      0:apagat      1:apagat      2:apagat      3:apagat
4:apagat    5:apagat      6:apagat

# Establir els nivells d'execució i llistar-los
[root@host ~]# chkconfig --level 345 xinetd on
[root@host ~]# chkconfig --list xinetd
xinetd      0:apagat      1:apagat      2:apagat      3:engegat
4:engegat   5:engegat      6:apagat
```

Cal comprovar si dins del xinetd el servei POP està actiu o no:

```
[root@host ~]# chkconfig --list | grep pop
ipop2:      apagat
ipop3:      engegat
pop3s:      apagat
```

• Monitorització (logs), lock i pid

El paquet del servidor pop no portava cap fitxer de gestió dels logs dins del *syslog* ni cap configuració pròpia de registre de logs. De totes maneres el servei pop funciona dins del superservei de xarxa *xinetd* i aquest sí que realitza una monitorització de les seves activitats.

Consultant l'ajuda de l'ordre *xinetd* i es poden observar els fitxers relacionats:

- el fitxer `xinetd.log(5)` descriu com funciona la monitorització dins del `xinetd`.
- el fitxer de configuració `xinetd.conf(5)` del `xinetd` també conté entrades que descriuen com ha de ser aquesta monitorització.

El següent fragment del fitxer de configuració permet observar que s'utilitza el dimoni `syslog` i es registren els events corresponents a les categories de daemon i info:

```
[root@host ~]# cat /etc/xinetd.conf | grep log
# Define general logging characteristics.
    log_type           = SYSLOG daemon info
    log_on_failure     = HOST
    log_on_success     = PID HOST DURATION EXIT
```

També podem observar que igual que els altres serveis “stand-alone” el superservei `xinetd` genera un fitxer de `lock` per indicar al sistema que està activat. La simple existència d'aquest fitxer (en realitat està buit) evita que es pugui iniciar més d'una vegada el servidor.

```
[root@host ~]# ll /var/lock/subsys/xinetd
-rw-r--r-- 1 root root 0 3 jun 17:55 /var/lock/subsys/xinetd
[root@host ~]# cat /var/lock/subsys/xinetd
```

Finalment observar el PID del procés del servei `xinetd`:

```
# Observar el pid del sevei xinetd
[root@host ~]# cat /var/log/messages | grep xinetd
Jun 3 18:55:47 portatil xinetd[4244]: START: tftp pid=4251 from=127.0.0.1

[root@host ~]# service xinetd status
xinetd (pid 3365) s'està executant...

[root@host ~]# ps ax | grep xinetd
3365 ?        Ss      0:00 xinetd -stayalive -pidfile /var/run/xinetd.pid

[root@host ~]# ll /var/run/xinetd.pid
-rw-r--r-- 1 root root 5 3 jun 17:55 /var/run/xinetd.pid
[root@host ~]# cat /var/run/xinetd.pid
3365
```

Com que el servidor POP (executable `/usr/sbin/ipop3d`) només s'activa durant una connexió pop caldrà estar atent per poder observar el seu PID. Per exemple Podem fer una connexió i mirar des d'una altra sessió els processos del sistema.

```
[root@portatil ~]# ps ax | grep pop
6391 ?        Ss      0:00 ipop3d
```

Instal·lació en un sistema Debian

1.5.3.Verificació del funcionament del servei POP.

Per comprovar el funcionament del servidor POP per a usuaris locals cal disposar d'un MUA que permeti l'accés remot als comptes de correu dels usuaris locals. Cal configurar un client de correu com *Thunderbird*, *mutt* o qualsevol altre per connectar-se al servidor POP local. També es pot usar una sessió *Telnet* o desenvolupar una aplicació en Python (per exemple) per connectar amb el servidor.

Com que el protocol POP és un protocol basat en TCP en la capa de transport, podem usar un *telnet* al port 110 d'un servidor de correu per simular una sessió POP.

Exemple 01: Sessió a un servidor local (1)

En la següent sessió es pot observar una connexió al servidor de correu POP local instal·lat anteriorment. Podeu obtenir una captura d'aquest diàleg feta amb *Wireshark* en el fitxer del material complementari:

- Complementari_AX_ud2_na3_pop_dialegLocal1.cap.

```
[root@host ~]# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
```

!!

Podeu analitzar tot el diàleg POP consultant el fitxer de captura del *Wireshark* proporcionat al material complementari.

```

: Escape character is '^]'.
: +OK POP3 localhost 2007a.104 server ready
USER pere
: +OK User name accepted, password please
PASS pere
: +OK Mailbox open, 1 messages
STAT
: +OK 1 480
NOOP
: +OK No-op to you too!
LIST
: +OK Mailbox scan listing follows
1 480
.
LIST 1
: +OK 1 480
RETR 1
: +OK 480 octets
Return-Path: <root@localhost.localdomain>
Received: from tftp.server.cat (localhost [127.0.0.1])
        by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DI4ig8005681
        for pere@tftp.server.cat; Fri, 13 Jun 2008 20:06:12 +0200
Date: Fri, 13 Jun 2008 20:04:44 +0200
From: root <root@localhost.localdomain>
Message-Id: <200806131806.m5DI4ig8005681@tftp.server.cat>
Status:

: aquest es un email qualsevol
: el text s'escrui fins acabar
: amb una linia que només conté un punt
.
QUIT
: +OK Sayonara
: Connection closed by foreign host.

```

Si fem el seguiment d'aquest diàleg podem observar que:

- Primerament s'entra en l'estat Authorization i un cop validat l'usuari amb les ordres USER i PASS s'entra en l'estat Transaction.
- L'ordre STAT indica que només existeix un missatge que ocupa 480 bytes.
- L'ordre NOOP obliga al servidor a contestar per validar que la connexió TCP es manté activa.
- L'ordre LIST sense arguments llista el número de missatge i els bytes que ocupa. Si s'indica un número de missatge ho indica només d'aquest missatge.
- Per recuperar o descarregar un missatge cal l'ordre RETR i el número de missatge.
- Per tancar una sessió POP s'utilitza la ordre QUIT.

Exemple 02: Sessió a un servidor local (2)

Aquesta és una altra sessió local a un servidor POP per l'accés remot al correu de l'usuari anna. Podeu obtenir una captura d'aquest diàleg feta amb wireshark en el fitxer del material complementari:

!!

Podeu analitzar tot el diàleg POP d'aquest segon exemple consultant el fitxer de captura del wireshark proporcionat al material complementari.

- Complementari_AX_ud2_na3_pop_dialogLocal2.cap

```

[root@portatil ~]# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
: +OK POP3 localhost 2007a.104 server ready
USER anna
: +OK User name accepted, password please
PASS anna
: +OK Mailbox open, 4 messages
STAT
: +OK 4 2434
LIST
: +OK Mailbox scan listing follows
1 616
2 607
3 596
4 615
.
LIST 3
: +OK 3 596
RETR 3
: +OK 596 octets
Return-Path: <pere@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
        by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DJZ7IG006648
        for <anna@tftp.server.cat>; Fri, 13 Jun 2008 21:35:10 +0200
Received: (from pere@localhost)
        by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DJZ7tt006647
        for anna; Fri, 13 Jun 2008 21:35:07 +0200
Date: Fri, 13 Jun 2008 21:35:07 +0200
From: pere@tftp.server.cat
Message-Id: <200806131935.m5DJZ7tt006647@tftp.server.cat>
To: anna@tftp.server.cat
Subject: missatge segon
Status:

: aquest es el segon missatge a l'anna.
adeu
.

```

```

:
:DELE 3
:+OK Message deleted
:LIST
:+OK Mailbox scan listing follows
:1 616
:2 607
:4 615
:
:
:TOP 4 1
:+OK Top of message follows
:Return-Path: <root@tftp.server.cat>
:Received: from tftp.server.cat (localhost [127.0.0.1])
:    by tftp.server.cat (8.14.1/8.14.1) with SMTP id m5DJZeeX006657
:    for <anna@tftp.server.cat>; Fri, 13 Jun 2008 21:35:41 +0200
:Received: (from root@localhost)
:    by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DJZdmu006656
:    for anna; Fri, 13 Jun 2008 21:35:39 +0200
:Date: Fri, 13 Jun 2008 21:35:39 +0200
:From: root <root@tftp.server.cat>
:Message-Id: <200806131935.m5DJZdmu006656@tftp.server.cat>
:To: anna@tftp.server.cat
:Subject: missatge de root a l'anna
>Status:
:
:aquest es un missatge de root a l'anna
:
:UIDL 4
:+OK 4 4852cc1300000004
:
:QUIT
:+OK Sayonara
:Connection closed by foreign host.
:

```

Si es segueix el diàleg efectuat es pot observar que:

- El client s'identifica amb les ordres USER i PASS. L'ordre STAT indica que hi ha 4 missatges que ocupen 2434 bytes tal i com es pot veure en fer el llista amb LIST.
- Es descarrega un missatge usant l'ordre RETR i el número de missatge. Un cop descarregat el missatge continua en el servidor, cal esborrar-lo si ja no es vol.
- El missatge s'esborra amb la ordre DELE i nº de missatge. En llistar a continuació s'observa que el missatge ja no hi és. De fet però, el missatge encara no s'ha esborrat (per això no estan numerats del 1 al tres els que queden), el missatge no s'esborrarà fins que es tanqui la sessió amb l'ordre QUIT i el servidor entri en l'estat d'actualització. Un cop s'ha marcat per esborrar un missatge es pot anular amb la ordre RSET.
- L'ordre TOP llista d'un missatge concret les seves capçaleres i les n primeres línies.
- L'ordre UIDL mostra l'identificador d'un missatge dins del servidor POP.

Exemple 03: Sessió a un servidor local (3)

Podem comprovar el funcionament del mecanisme d'esborrar els missatges amb el següent diàleg. Podeu obtenir una captura d'aquest diàleg feta amb wireshark en el fitxer del material complementari:

- Complementari_AX_ud2_na3_pop_dialogLocal3.cap

```

:[root@portatil ~]# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK POP3 localhost 2007a.104 server ready
USER anna
+OK User name accepted, password please
PASS anna
+OK Mailbox open, 3 messages
:LIST
:+OK Mailbox scan listing follows
:1 616
:2 607
:3 615
:
:DELE 1
:+OK Message deleted
:DELE 2
:+OK Message deleted
:LIST
:+OK Mailbox scan listing follows
:3 615
:
:RSET
:+OK Reset state
:LIST
:+OK Mailbox scan listing follows
:1 616
:2 607
:3 615
:
:QUIT
:+OK Sayonara
:

```

!!

Podeu analitzar tot el diàleg POP d'aquest tercer exemple consultant el fitxer de captura del wireshark proporcionat al material complementari.

```
Connection closed by foreign host.
```

En el diàleg anterior es pot observar que s'han eliminat els missatges 1 i 2 amb l'ordre *DELE*, però posteriorment s'ha realitzat l'ordre *RSET* que anula els canvis fets en els missatges i els deixa com estaven. Per tant els missatges 1 i 2 no s'han esborrat tal i com s'observa de la ordre *LIST* que fa un llistat dels números de missatges (hi ha els mateixos que en començar).

Exemple 04: Sessió a un servidor remot

Utilitzant la utilitat *pop3test* es pot fer una sessió a un compte de *Gmail*, tenint en compte que cal indicar l'usuari per a la autenticació, també per a l'autorització i indicar que la comunicació és SSL. Podeu obtenir una captura d'aquest diàleg feta amb wireshark en el fitxer del material complementari:

- Complementari_AX_ud2_na3_pop_dialogGmail.cap

!!

Podeu analitzar tot el diàleg POP d'aquest quart exemple consultant el fitxer de captura del *wireshark* proporcionat al material complementari.

```
[root@host ~]# pop3test -u ecanetlprofe -a ecanetlprofe -s pop.gmail.com
verify error:num=20:unable to get local issuer certificate
verify error:num=27:certificate not trusted
verify error:num=21:unable to verify the first certificate
TLS connection established: TLSv1 with cipher RC4-MD5 (128/128 bits)
S: +OK Gpop ready for requests from 81.34.247.52 u14pf4353460gvf.0
C: CAPA
S: +OK Capability list follows
S: USER
S: RESP-CODES
S: EXPIRE 0
S: LOGIN-DELAY 300
S: X-GOOGLE-VERHOEVEN
S: UIDL
S: .
Please enter your password:
C: USER ecanetlprofe
S: +OK send PASS
C: PASS <omitted>
S: +OK Welcome.
Authenticated.
Security strength factor: 128
LIST
+OK 5 messages (25230 bytes)
1 2641
2 1996
3 2022
4 17169
5 1402
.
LIST 4
+OK 4 17169
TOP 4 3
+OK message follows
Received: by 10.82.154.13 with HTTP; Fri, 13 Jun 2008 09:31:06 -0700 (PDT)
Message-ID: <34af42060806130931n46458220rae640f34f5e4bfb2@mail.gmail.com>
Date: Fri, 13 Jun 2008 18:31:06 +0200
From: "eduard canet" <ecanetlprofe@gmail.com>
To: ecanetlprofe@gmail.com
Subject: missatge de webmail amb contingut adjunt
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="-----_Part_476_8183242.1213374666169"
Delivered-To: ecanetlprofe@gmail.com

-----=_Part_476_8183242.1213374666169
Content-Type: multipart/alternative;
        boundary="-----_Part_477_4440518.1213374666170"
.

RETR 4
+OK message follows
Received: by 10.82.154.13 with HTTP; Fri, 13 Jun 2008 09:31:06 -0700 (PDT)
Message-ID: <34af42060806130931n46458220rae640f34f5e4bfb2@mail.gmail.com>
Date: Fri, 13 Jun 2008 18:31:06 +0200
From: "eduard canet" <ecanetlprofe@gmail.com>
To: ecanetlprofe@gmail.com
Subject: missatge de webmail amb contingut adjunt
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="-----_Part_476_8183242.1213374666169"
Delivered-To: ecanetlprofe@gmail.com

-----=_Part_476_8183242.1213374666169
... output suprimir ...
-----=_Part_476_8183242.1213374666169--
.
QUIT
+OK Farewell.
Connection closed.
```

Monitoritzar el tràfic POP

Si s'utilitza Wireshark per monitoritzar el tràfic d'una connexió POP es podran observar els ports client (dinàmic) i del servidor (110), el tipus de tràfic TCP, la connexió TCP de tres vies, la petició GET del client i la resposta del servidor. En la Figura 8 "Pantalla de captura de wireshark d'un diàleg local POP" es pot observar el seguiment del diàleg POP en text pla. Podeu obtenir una captura del wireshark del fitxer del material complementari:

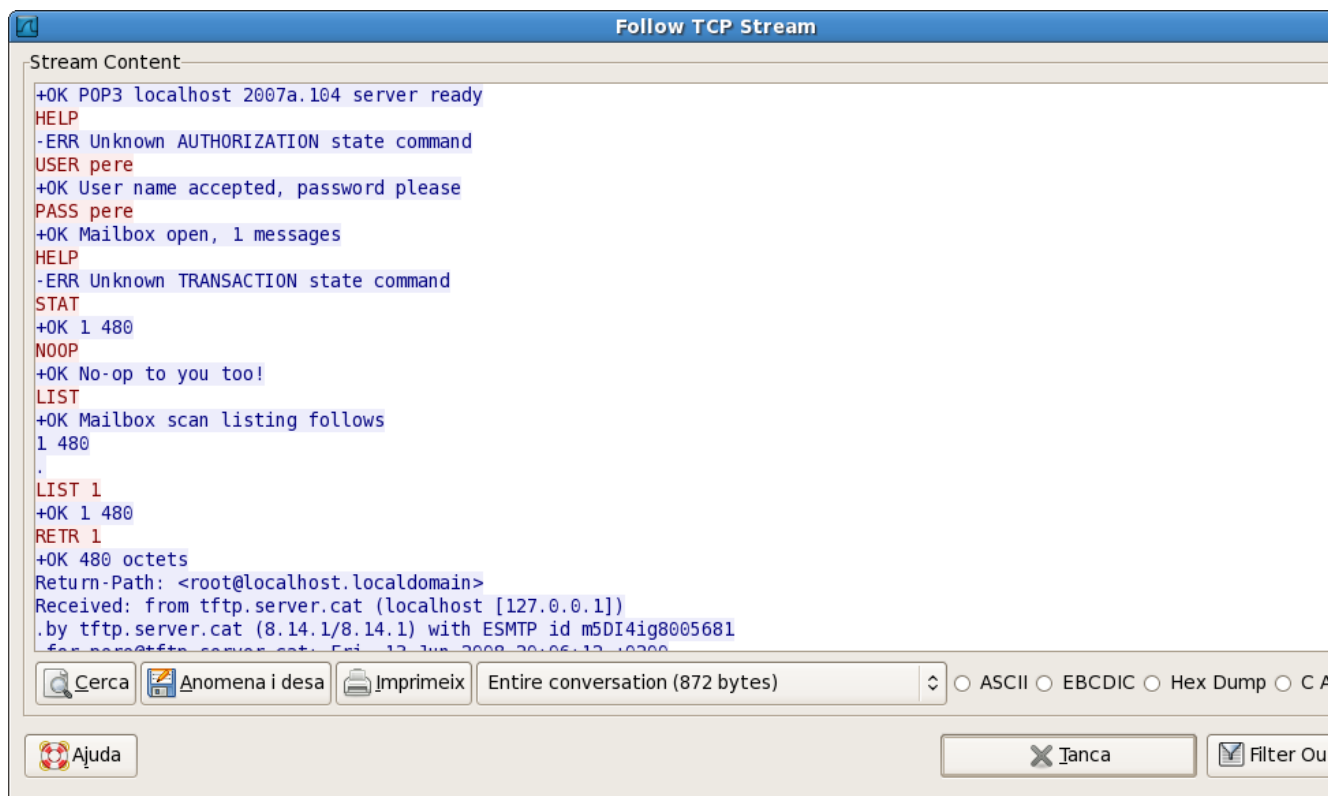
!!

Podeu analitzar tot el diàleg POP d'aquest exemple consultant el fitxer de captura del

- Complementari_AX_ud2_na3_pop_dialegLocal1.cap

wireshark proporcionat al material complementari.

Figura 8. Pantalla de captura de wireshark d'un diàleg local POP.



Un altre exemple de tràfic POP es pot realitzar entre un client de correu local com el *thunderbird* i un webmail com per exemple *Gmail*. La conversa amb el servidor POP de Gmail utilitzant SSL es pot consultar també amb el fitxer de material complementari:

- Complementari_AX_ud2_na3_pop_dialegGmail.cap

Client gràfic thunderbird: Servidor POP local

Existeixen multitud de clients de correu gràfics. Un dels més coneguts i utilitzats en GNU/Linux és el *thunderbird*, per a Windows un exemple de client de correu gràfic és el *Outlook*. Per configurar un compte de correu (o més d'un) en un MUA com el *thunderbird* cal disposar d'un compte de correu en un servidor de correu. Els elements més importants en la configuració són:

- Identificar clarament el compte de correu de l'usuari i la identitat associada.
- Indicar quin és el servidor de correu sortint. El correu sortint generalment és enviat per un MTA que utilitza el protocol SMTP. Sovint els administradors dels dominis han posat alias al servidor de correu de manera que s'identifica al servidor sortint amb noms del tipus *smtp.servidor.domini*.
- Per tal de recuperar el correu del servidor cal indicar el nom del servidor i el protocol a utilitzar per l'accés remot a la bústia de correu. El protocol usat generalment és POP o IMAP. Molts ISP han posat alias als servidors de manera que les màquines tenen noms del tipus *pop.servidor.domini* o *imap.servidor.domini*. Per accedir al correu cal identificar-se indicant el nom de l'usuari i sovint cal indicar una contrasenya.
- En entorns de comunicació que exigeixen privacitat de les dades cal configurar correctament el mecanisme de transport segur a utilitzar, SSL o TLS i configurar-lo correctament.

Sempre que es configura un compte de correu en un client de correu cal disposar de la següent informació:

- correu **SMTP sortint**: nom del servidor, port i aspectes de

seguretat i autenticació.

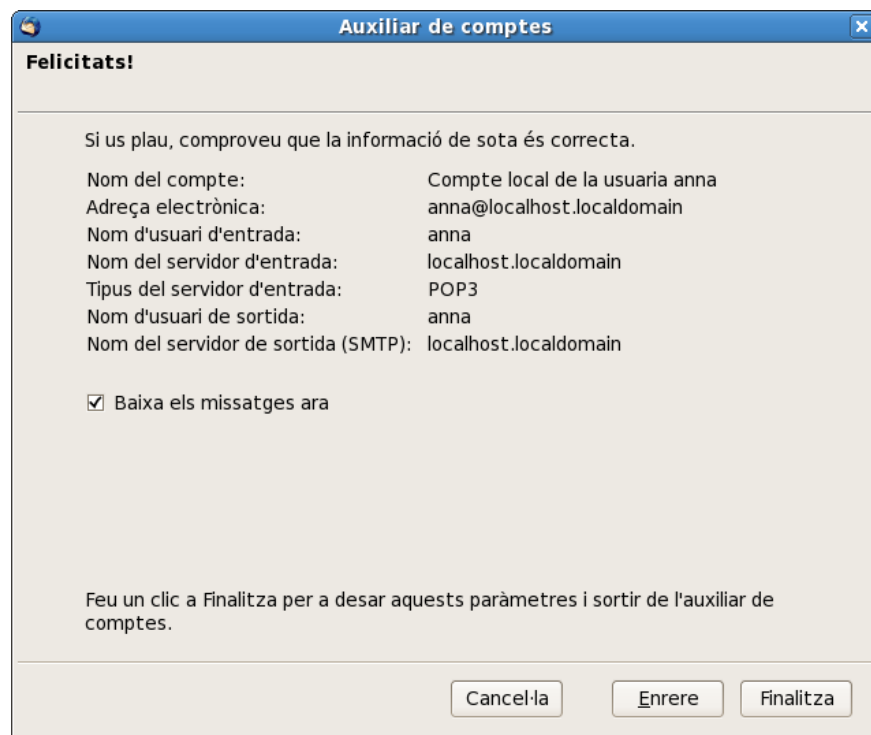
- correu **entrant (POP o IMAP)**: nom del servidor, port i aspectes de seguretat i autenticació.

Si es vol crear en el thunderbird un compte per poder accedir al correu de l'usuària "anna" en el servidor POP local, un repàs ràpid als passos a efectuar en el thunderbird són:

- seleccionar que es vol crear un nou compte de correu.
- indicar el nom de l'usuari ("anna") i el seu correu (anna@localhost.localdomain).
- indicar el nom del servidor de correu SMTP sortint i el del servidor de correu POP entrant (localhost.localdomain tots dos).
- indicar el nom amb el que s'identificarà aquest compte dins del thunderbird ("anna"). Si un usuari configura més d'un compte de correu al *thunderbird* convé que assigni a cada compte noms clarificadors, per exemple: "anna personal", "anna feina", "anna a gmail", etc.
- escriure una descripció del compte, per exemple "compte de prova per a l'usuària local anna".

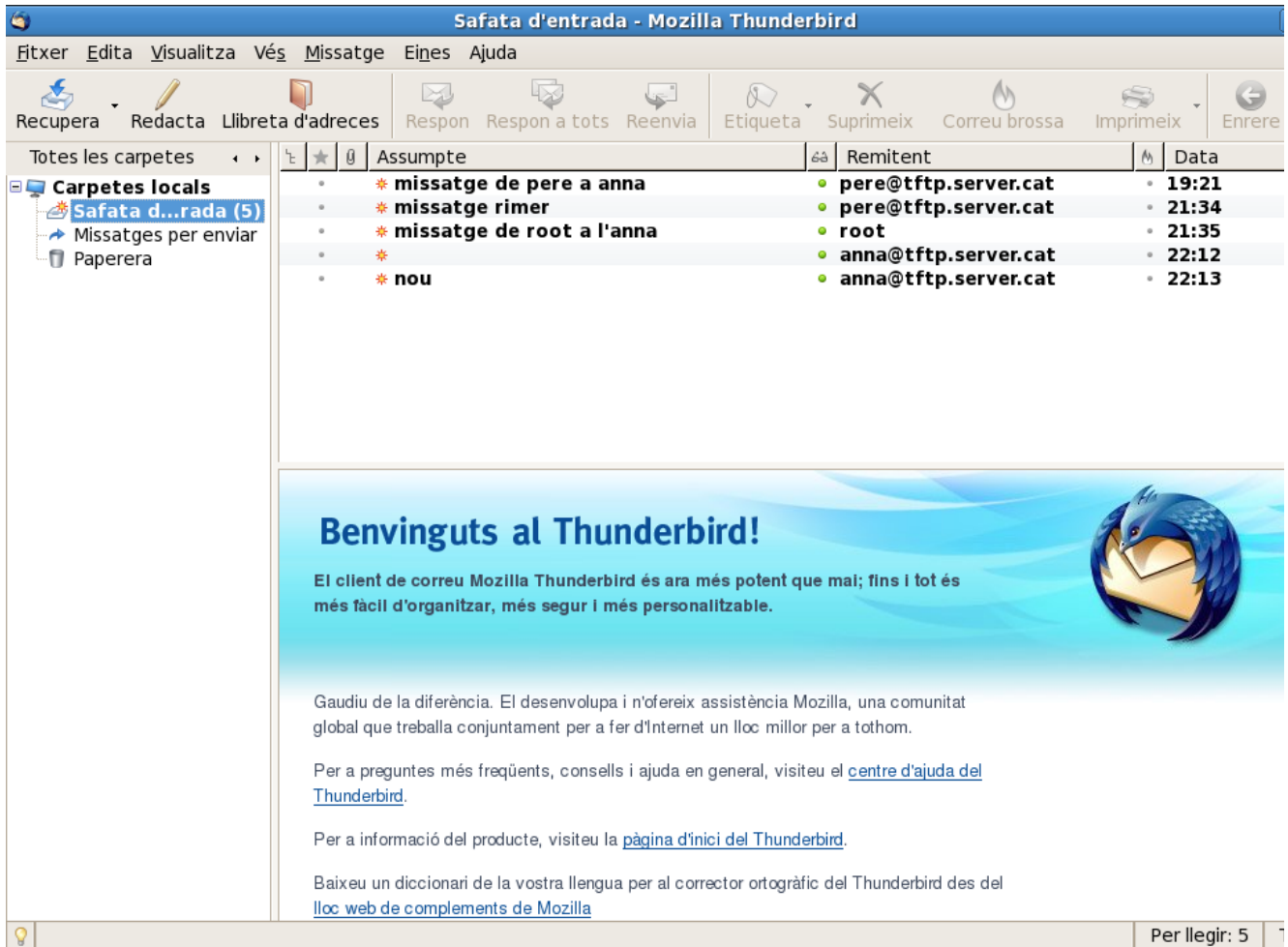
La Figura 9 "Panell resum de la creació d'un compte de correu local en el thunderbird" mostra la pantalla resum del *thunderbird* que es mostra en finalitzar la creació d'un compte de correu. En ella es pot observar l'informació necessària per configurar apropiadament un compte de correu.

Figura 9. Panell resum de la creació d'un compte de correu local en el thunderbird.



La Figura 10 "Pantalla del compte d'usuari POP en el thunderbird" mostra el compte d'un usuari local en el *thunderbird* que accedeix via POP al servidor local.

Figura 10. Pantalla del compte d'usuari POP en el thunderbird.



Client gràfic thunderbird: Servidor POP webmail

La majoria de servidors Webmail com *Gmail*, *Yahoo*, etc permeten descarregar-se el correu des d'un client POP o IMAP. Podem configurar Thunderbird per connectar amb el servidor de Gmail.

Primerament cal configurar *Gmail* per permetre fer això, cal anar a *settings* i a *forwarding POP & IMAP*. Cal assegurar-se de deixar activat que es permet la descàrrega per POP i/o per IMAP. Gmail proporciona instruccions de configuració de clients gràfics com thunderbird (entre altres) per descarregar el correu de gmail.

Un cop fet això es pot configurar el compte d'usuari de Gmail en el Thunderbird per descarregar-lo via POP:

- Indicar el nom propi de l'usuari i el compte de correu a usar (nom del compte complet, usuari @domini).
- Indicar el tipus d'accés remot que s'utilitzarà, POP o IMAP. Dir també el nom del servidor a utilitzar.
- Indicar el nom d'usuari del compte de correu.
- Escriure un identificador del compte de correu. Si voleu es pot deixar el propi nom del compte, si en teniu molts potser es millor escriure una frase descriptiva del tipus: compte de la feina, compte de casa, compte dels amics, compte per a delictes, etc.
- Un cop indicades totes les dades es mostra el resum.
- Un cop creat el compte apareix la pantalla de treball de Thunderbird. Encara cal configurar la seguretat.
- La comunicació amb Gmail ha de ser xifrada, per això cal activar SSL.
- En la configuració del servidor de correu sortint cal indicar que s'utilitzi TLS si en disposa.

La Figura 11 "Pantalla de configuració de Gmail" mostra la pantalla de configuració de Gmail que permet configurar un compte d'usuari per descarregar-se remotament el correu via POP o IMAP.

Figura 11. Pantalla de configuració de Gmail.

Gmail - Settings - ecanet1profe@gmail.com - Mozilla Firefox

Fitxer Edita Visualitza Historial Adreces d'interès Eines Ajuda

http://mail.google.com/mail/#settings/fwdandpop

Gmail Calendar Documents Photos Reader Web more ▼ ecanet1profe@gmail.com

Compose Mail


[Inbox](#)
[Starred](#) ★
[Chats](#)
[Sent Mail](#)
[Drafts](#)
[All Mail](#)
[Spam](#)
[Trash](#)

Contacts

▼ Chat

Search, add, or invite

● eduard canet
Set status here ▼

 Chat with your AIM® buddies
[Get started](#)

● Eduard Canet
CARME GARCIA PI
jlope295
mgonz445
mmarti80
[Options](#) [Add Contact](#)

▼ Labels

[cueta](#) ☐
[ICE_2007](#) ☐
[old_ice_2006](#) ☐
[Edit labels](#)

▼ Invite a friend

Give Gmail to:

Send Invite 50 left
[Preview Invite](#)

Settings

[General](#) [Accounts](#) [Labels](#) [Filters](#) **Forwarding and POP/IMAP** [Chat](#) [Web Clips](#) [Labs](#)

Forwarding:

☒ Disable forwarding
☐ Forward a copy of incoming mail to and keep Gmail's copy in

Tip: You can also forward only some of your mail by [creating a filter](#)

POP Download:
[Learn more](#)

1. Status: POP is enabled for all mail that has arrived since 7/6/07
☐ Enable POP for **all mail** (even mail that's already been downloaded)
☐ Enable POP for **mail that arrives from now on**
☐ **Disable POP**

2. When messages are accessed with POP keep Gmail's copy in the Inbox

3. Configure your email client (e.g. Outlook, Eudora, Netscape Mail)
[Configuration instructions](#)

IMAP Access:
(access Gmail from other clients using IMAP)
[Learn more](#)

1. Status: IMAP is enabled
☒ Enable IMAP
☐ Disable IMAP

2. Configure your email client (e.g. Outlook, Thunderbird, iPhone)
[Configuration instructions](#)

[Import contacts](#) from Yahoo, Outlook, and others into your Gmail contact list. [Learn more](#)

You are currently using 0 MB (0%) of your 6822 MB.
 Gmail view: standard | [turn off chat](#) | [basic HTML](#) | [Learn more](#)

©2008 Google - [Terms](#) - [Google Home](#)

Fet

1.5.4.El servei IMAP.

IMAP (Internet message access protocol o protocol d'accés a missatges d'Internet) és un protocol de capa d'aplicació del model TCP/IP que proporciona a l'usuari accés remot a la seva bústia de correu. L'IMAP sorgeix com a resposta al problema d'accedir al correu des de diferents ordinadors utilitzant POP.

El POP és un protocol pensat per baixar el correu del servidor al PC local de l'usuari i poder-lo manipular després sense connexió a Internet. Usant POP es considera que el correu resideix en l'equip de l'usuari, que baixa tot el correu de cop cada vegada que es connecta al servidor de correu. Quan els usuaris es van acostumar a consultar el correu remotament, ho van començar a fer des d'equips diferents: a casa, a la feina, de vacances, etc. Cada cop que ho feien deixaven part del seu correu en llocs diferents. El que s'ha baixat a casa no es pot consultar a la feina, etc. Un cop els usuaris es van acostumar a disposar de connexió d'Internet més assiduament, calia un mecanisme més evolucionat d'accés remot al correu.

L'IMAP fa un enfocament diferent, els missatges de correu es dipositen en el servidor, i és en el servidor on s'emmagatzemen estructurats en carpetes (o *folders* o *mailbox*) i on es manipulen. L'usuari els pot baixar localment, però com a còpia temporal. Per tant, tota la gestió dels missatges de correu té lloc en el servidor. Això fa de l'IMAP un protocol més complex que el POP.

Ni l'IMAP ni el POP són protocols de transmissió de correu. Usualment és el protocol SMTP qui fa aquesta funció.

!!

L'**IMAP** és un protocol de capa d'aplicació de la pila de protocols TCP/IP (port 143) definit en el document RFC 1064. Permet a un client de correu (MUA) obtenir remotament el correu dipositat en la bústia de correu de l'usuari en un servidor de correu IMAP.

Per obtenir més informació sobre l'especificació del protocol IMAP en els RFC 1064 i 3501, aneu a la secció "Adreces d'interès" del web d'aquest crèdit.

L'IMAP sorgeix al 1986 amb el nom d'*interim mail acces protocol*, que en la versió següent es canvia per *interactive mail access protocol* (document RFC 1064), que posteriorment es canviarà per *Internet mail acces protocol*. L'evolució actual és IMAP versió 4 revisió 1 (març 2003) corresponent al document RFC 3501 (del qual també s'han fet actualitzacions i extensions) que s'ha creat sota els auspicis de l'IETF.

El protocol IMAP està pensat per tenir en el servidor tot el correu de l'usuari organitzat en carpetes jeràrquiques de manera indefinida. Es permet la manipulació remota de les carpetes i els missatges. Es poden crear, modificar i suprimir carpetes i missatges. Els missatges no s'esborren si no ho indica explícitament l'usuari. A més a més, aporta la funcionalitat de cerca i filtratge de missatges directament en el servidor. És a dir, no cal baixar els missatges per buscar-ne els que compleixen unes condicions determinades. Permet l'accés concurrent de diversos usuaris a la mateixa bústia, i el servidor pot notificar l'arribada de correu nou. Els missatges multipart es poden baixar parcialment, es poden buscar parts i baixar només les que interessin. Tots els missatges i les bústies tenen indicadors d'estat que descriuen, per exemple, si el missatge s'ha llegit, contestat, si és nou, etc.

Difusió del servei IMAP

Avui en dia l'IMAP està molt estès, però no és estrany trobar servidors ISP i portals de correu web que permeten baixar el correu únicament mitjançant el POP.

El model IMAP

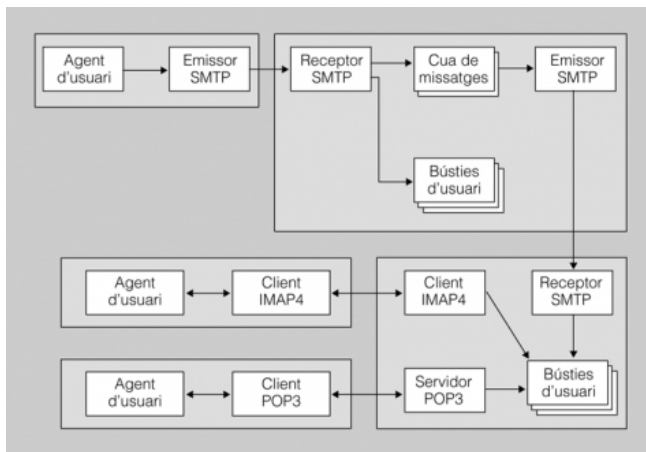
Igual que la majoria de serveis de xarxa a Internet, el protocol IMAP s'estructura seguint l'esquema client/servidor. Aquests són els agents que intervien en una comunicació IMAP:

- **MUA (mail user agent o agent d'usuari de correu).** L'usuari interactua amb un agent d'usuari per accedir al correu mitjançant l'IMAP. Són agents d'usuari programes com el Thunderbird, Getmail, Fetchmail, MS Outlook Express, Eudora, Gmail, etc. Com es pot veure en la llista de les aplicacions MUA, poden ser de text, gràfiques i, fins i tot, pel web. Aquestes aplicacions incorporen el programari necessari per actuar com a clients IMAP.
- **Client IMAP.** La part pròpiament encarregada de comunicar amb el servidor IMAP per obtenir els missatges de la bústia de correu de l'usuari és el client IMAP. Client i servidor IMAP parlen un llenguatge comú que és el protocol IMAP.
- **Servidor IMAP.** Per implementar l'accés remot al correu cal disposar d'un servidor IMAP en funcionament. Aquest servidor IMAP conté les bústies dels usuaris o hi accedeix. Els missatges es reben per SMTP i és un MTA o un MDA qui els diposita en la bústia. El servidor IMAP

atén les peticions dels clients IMAP per gestionar el correu.

Es pot observar el model funcional del protocol IMAP en la Figura 12 "Model funcional del protocol IMAP", on es veu que un usuari mitjançant el seu MUA es baixa el correu del servidor amb el POP o IMAP.

Figura 12. Model funcional del protocol IMAP.



En el protocol IMAP es detallen quatre estats clarament diferenciats:

1) No autenticat . Quan s'estableix la connexió TCP/IP entre el client i el servidor s'entra en aquest estat. El client s'ha d'autenticar davant del servidor, acreditant ser un usuari vàlid. Per fer-ho indicarà el nom d'usuari i la contrasenya.

2) Autenticat . Un cop autenticat i abans de poder manipular missatges, ha de seleccionar la carpeta (o bústia o *folder* o *mailbox*) amb la qual operarà. En aquest estat pot manipular les carpetes (crear, esborrar, modificar i veure l'estat) però no els missatges fins que no se n'ha seleccionada una.

3) Seleccionat . Un cop s'ha seleccionat una carpeta amb èxit, s'entra en aquest estat, que permet la manipulació dels continguts de la carpeta.

4) Logout . En aquest estat es procedeix a tancar la connexió. S'hi pot arribar tant per petició del client com per decisió unilateral del servidor.

El servidor IMAP emmagatzema permanentment tots els missatges de l'usuari. Per fer-ho utilitza un sistema de carpetes (o bústies) jeràrquiques i atributs que descriuen tant l'estat de les bústies com dels missatges:

- **Carpetes (mailbox)**. Hi ha una bústia o *mailbox* que és la de l'usuari. Dins d'aquesta bústia, s'hi poden crear carpetes que s'indiquen de manera relativa igual que amb els directoris d'una estructura de fitxers. Les carpetes disposen almenys de dos atributs:
- **Next UID (UID següent)**. Indica el valor UID que s'assignarà al missatge següent que arribi.
- **UID Validity Value (UIDVALIDITY)**. És un valor d'identificador únic assignat a la carpeta seleccionada. La combinació de nom de carpeta UIDVALIDITY i UID identifiquen de manera perpètua un missatge en el servidor.

Atributs de missatge. Els missatges tenen atributs que s'emmagatzemen en les pròpies bústies que en faciliten la gestió:

- **UID**. Identificador únic del missatge. És un número de 32 bits que s'assigna ascendentment a mesura que arriben missatges (no necessàriament correlatiu). Això permet al servidor saber, en una bústia, a partir de quin número de missatge hi ha els missatges nous (en POP això no és possible)
- **Número de seqüència**. Número de seqüència relatiu del missatge dins de la bústia (de 1 a n per a n missatges). Els números de seqüència s'assignen correlativament segons el UID en ordre ascendent i varien en esborrar-se i afegir-se nous missatges.
- **Indicadors**. Els indicadors o *flags* (banderes) informen de l'estat del missatge. Per exemple, si s'ha llegit, esborrat, etc. Els indicadors són: *Seen* (llegit), *Answered* (respost), *Flagged* (marcat), *Deleted* (esborrat), *Draft* (esborrany) i *Recent* (nou).

- **Data interna.** Data i hora d'arribada del missatge al servidor IMAP (no és la data i hora de l'emissió del missatge que hi ha en la capçalera *Date*).
- **Longitud.** Nombre de bytes del missatge.
- **Estructura del sobre.** Representació analitzada de les capçaleres del missatge.
- **Estructura del cos.** Representació analitzada de l'estructura MIME del cos del missatge.
- **Parts de text del missatge.** Per permetre la cerca de les diferents parts de text del missatge. Es pot fer l'accés segons la part de capçaleres, cos, part del cos MIME i capçalera MIME.

Funcionament IMAP

El funcionament generalitzat del protocol IMAP és que el client fa una connexió TCP/IP al port 143 del servidor i s'inicia un diàleg entre el client i el servidor, en què tant el client com el servidor poden prendre la iniciativa. En aquest procés se succeeixen els quatre estats del protocol IMAP (no autènticat, autènticat, seleccionat i *logout*) i s'intercanvien ordres i respostes seguint el model de diàleg de Telnet:

- **Ordres.** Les ordres IMAP són ordres de text que inclouen un *tag* inicial (o etiqueta curta), l'ordre i els seus arguments. Cada ordre comença amb una etiqueta inicial diferent per diferenciar-la de les altres ordres. Quan el servidor emeti la resposta final de l'ordre i indiqui si s'ha realitzat correctament o no, ho farà mostrant l'etiqueta de l'ordre a la qual respon. Per exemple, es pot usar *a001* per a la primera ordre, *a002* per a la segona, etc. El client pot enviar ordres sense esperar a que finalitzin les precedents.
- **Respostes.** El servidor pot enviar dades al client tant com a resposta a una ordre com de manera unilateral (per exemple, per informar que hi ha correu nou). El client ha d'estar en tot moment a punt per rebre aquestes dades. Que el servidor envii dades al client no significa que l'execució de l'ordre del client hagi finalitzat. Només es dona per finalitzada quan el servidor emet una resposta amb la mateixa etiqueta que l'ordre del client. El servidor pot processar una ordre abans d'acabar de processar l'ordre anterior.

Tot seguit es fa una llista de les ordres utilitzables en el protocol IMAP agrupades segons l'estat:

1) Qualsevol estat (ordres generals)

- **Capability.** Llista les capacitats del servidor. Permet al client saber quines són les prestacions del servidor.
- **Noop.** No fa res, exigeix resposta afirmativa del servidor. Permet al client saber si encara es manté la connexió.
- **Logout.** Notificació del client al servidor per fer-li saber que vol finalitzar la connexió.

2) No autènticat

- **Authenticate tipus.** Indica al servidor el mecanisme d'autenticació a utilitzar.
- **Login user password.** El client s'identifica davant del servidor indicant el *login* i la contrasenya. El format variarà (text net, *hash* MD5, etc.) segons el tipus d'autenticació utilitzat.

3) Autènticat

- **Select bústia.** Selecciona la bústia amb què ha d'operar. En fer-ho, el servidor emet una resposta no etiquetada en què informa dels atributs (*flags*) de la bústia, del nombre de missatges que conté (*exists*) i del nombre de missatges recents (*recent*). També pot indicar el número del primer missatge no llegit (*nseen*) i la llista *deflags* que es poden modificar (*permanentflags*).
- **Examine bústia.** Realitza la mateixa funció que l'ordre *Select* però només de lectura.

Diferència entre POP i IMAP

En el protocol POP, el servidor només pot respondre a peticions del client, però no pot prendre la iniciativa. En el protocol IMAP sí.

- **Create bústia.** Crea la bústia amb el nom indicat.
- **Delete bústia.** Esborra la bústia indicada.
- **Rename bústia nomNou.** Permet modificar el nom de la bústia assignant-li un nom nou.
- **Subscribe bústia.** Les bústies poden estar actives o no actives. Les bústies s'activen amb l'ordre *Subscribe*.
- **Unsubscribe bústia.** Permet desactivar una bústia.
- **List bústia criteri.** Llista les bústies que compleixen el criteri indicat dins de la bústia seleccionada.
- **Lsub bústia criteri.** Realitza la mateixa funció que l'ordre anterior però només per a les bústies actives.
- **Status bústia atributs.** Permet conèixer l'estat d'una bústia per mitjà dels seus atributs. Els atributs són els següents:
 - *MESSAGE*: nombre de missatges dins la bústia;
 - *RECENT*: nombre de missatges recents (nous);
 - *UIDNEXT*: UID que s'assignarà al missatge nou següent que arribi a la bústia;
 - *UIDVALIDITY*: valor de UID de la bústia;
 - *UNSEEN*: nombre de missatges no llegits.
- **Append bústia [atributs] [data-hora] literal.** Permet afegir un text literal al final de la bústia com si es tractés d'un missatge nou. El missatge s'afegeix amb la data, hora i atributs indicats.

Avantatge de l'IMAP respecte al POP

Un dels avantatges de l'IMAP respecte al POP és que el servidor sap a partir de quin número de missatge hi ha els missatges nous (*recent*).

4) Selecció

- **Check.** El client sol·licita al servidor que es faci un punt de control de la bústia en un moment determinat.
- **Close.** Tanca la bústia i elimina tots els missatges que conté que tenen l'indicador d'esborrament (*deleted*) activat.
- **Expunge.** Permet esborrar els missatges que tenen l'atribut d'esborrament (*deleted*) activat sense que calgui tancar la bústia.
- **Search [charset] criteri.** Permet buscar missatges dins de la bústia que compleixen el criteri de cerca indicat.
- **Fetch dades atributsRecuperació.** Permet recuperar un conjunt de missatges totalment o parcialment segons els atributs de recuperació indicats.
- **Store conjuntMissatges atributs.** Permet alterar les dades d'atributs associats a un conjunt de missatges en una bústia.
- **Copy conjuntMissatges bústia.** Copia un conjunt de missatges al final de la bústia indicada.
- **UID ordre.** Retorna l'UID d'un missatge. S'utilitza conjuntament amb les ordres COPY, FETCH, STORE o SEARCH per retornar els UID manipulats.

5) Experimental

- **X<ordre>.** Es poden desenvolupar ordres experimentals fora de l'especificació IMAP. Per fer-ho cal que les ordres comencin amb *XnomComanda*. D'aquesta manera s'evita que es produeixin conflictes amb ordres futures.

En l'exemple següent es pot veure tot el diàleg client/servidor d'una sessió IMAP utilitzant Telnet.

```
[root@portatil ~]# telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4REV1 LITERAL+ SASL-IR LOGIN-REFERRALS
STARTTLS] localhost IMAP4rev1 2007a.403 at Sat, 14 Jun 2008
13:16:47 +0200 (CEST)

a003 LOGIN pere pere
a003 OK [CAPABILITY IMAP4REV1 LITERAL+ IDLE UIDPLUS NAMESPACE
CHILDREN MAILBOX-REFERRALS BINARY UNSELECT ESEARCH WITHIN SCAN
SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND] User
pere authenticated
```

```

a004 SELECT inbox
* 4 EXISTS
* NO Mailbox vulnerable - directory /var/spool/mail must have
1777 protection
* 4 RECENT
* OK [UIDVALIDITY 1213385060] UID validity status
* OK [UIDNEXT 6] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft
\Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/pere
* NO Mailbox vulnerable - directory /var/spool/mail must have
1777 protection
a004 OK [READ-WRITE] SELECT completed

a005 FETCH 1 rfc822.text
* 1 FETCH (RFC822.TEXT {63}
exemple de missatge de la usuaria anna
a l'usuari pere
adeu
)
a005 OK FETCH completed

a006 LOGOUT
* BYE portàtil IMAP4rev1 server terminating connection
a006 OK LOGOUT completed

```

1.5.5. Instal·lació del servei IMAP.

Tot seguit es descriurà el procés per instal·lar el servei IMAP en un entorn GNU/Linux. Un cop feta la instal·lació cal observar què s'ha instal·lat, quins programes executables, on són els fitxers de configuració, els de monitoratge, etc.

El procés és mimètic al descrit per instal·lar el servidor SMTP. Per treballar amb un servidor de correu POP i IMAP s'instal·larà el software *uw-imap* que proporciona totes dues funcionalitats. Es tracta del servidor elaborat per la Universitat de Washington.

La principal diferència entre aquest servidor *uw-imap* i els instal·lats anteriorment és que es tracta d'un servei de xarxa configurat per executar-se dins del super servei de xarxa *xinetd*.

Instal·lació en un sistema Fedora

El procés per instal·lar el software de servidor IMAP usant el servidor *uw-imap* és el mateix que s'ha utilitzat per al servei POP. Per tant es recomana seguir els passos realitzats en aquell apartat.

En resum s'observa que:

- L'executable del servei és el fitxer */usr/sbin/imapd*. S'identifica perquè pertany a un directori d'executables i acaba amb la lletra "d" de daemon.
- La configuració del servei es troba en el fitxer */etc/xinetd.d/imap* i */etc/xinetd.d/imapd*. El fitxer de configuració del servei és un fitxer de configuració dins del superservei de xarxa *xinetd*. N'existeix un per el servei IMAP que realitza tràfic en text pla i un pel servei IMAPS que utilitza transport segur basat en SSL i TLS.
- Tipus de servei: El servidor *uw-imap* actua dins del superservei de xarxa *xinetd*. El *xinetd* s'executa i escolta les peticions IMAP entrants i en rebre-les engega el servidor *uw-imap*.

Per engegar el servei i configurar que estigui actiu per defecte el els *runlevels* desitjats cal seguir els següents passos:

- Editar el fitxer que configura el servei *imap* dins del servei *xinetd*. Cal modificar la línia que diu "*disable=yes*" i canviar aquest valor per "*no*". Aquesta acció cal fer-la tant per el servei *imap* com pel servei segur *imaps*.
- Establir els nivells per defecte en què s'ha d'executar el super servei de xarxa *xinetd*.
- Comprovar que el servei *xinetd* es troba activat en el *runlevel* actual i verificar que també ho estan els serveis de *imap*.

```
# Configuració d'imap dins del superservei xinetd
```

!!

El procés d'instal·lació del servidor està descrit en l'apartat "Instal·lació del servidor POP."

```
[root@host ~]# cat /etc/xinetd.d/imap
# default: off
# description: The IMAP service allows remote users to access their mail \
# using an IMAP client such as Mutt, Pine, fetchmail, or Netscape \
# Communicator.
service imap
{
    socket_type      = stream
    wait             = no
    user             = root
    server           = /usr/sbin/imapd
    log_on_success   += HOST DURATION
    log_on_failure   += HOST
    disable          = no
}

# Comprovar que el super servei de xarxa està actiu
[root@host ~]# /etc/rc.d/init.d/xinetd start
S'està iniciant el xinetd:

# Comprovar si dins del xinetd el servei IMAP està actiu o no:
[root@host ~]# chkconfig --list | grep imap
imap:          engegat
imaps:         apagat
```

Un cop instal·lat el software del servidor cal saber identificar com es realitza la monitorització o logs del servei, el lock i identificar-ne el pid. Tots aquests passos són idèntics als descrits en la instal·lació del servidor POP.

Instal·lació en un sistema Debian

1.5.6.Verificació del funcionament del servei IMAP.

Per comprovar el funcionament del servidor IMAP per a usuaris locals cal disposar d'un MUA que permeti l'accés remot als comptes de correu dels usuaris locals. Cal configurar un client de correu com *Thunderbird*, *mutt* o qualsevol altre per connectar-se al servidor IMAP local. També es pot usar una sessió *Telnet* o desenvolupar una aplicació en Python (per exemple) per connectar amb el servidor.

Com que el protocol IMAP és un protocol basat en TCP en la capa de transport, podem usar un *telnet* al port 143 d'un servidor de correu per simular una sessió IMAP.

!!

Es pot fer un seguiment i anàlisi de tot el diàleg d'aquest exemple consultant el fitxer de captura de *wireshark* del material complementari.

Exemple 01: Sessió a un servidor local (1)

En la següent sessió es pot observar una connexió al servidor de correu IMAP local instal·lat anteriorment. Podeu obtenir una captura d'aquest diàleg feta amb *wireshark* en el fitxer del material complementari:

- Complementari_AX_ud2_na3_imap_dialegLocal1.cap

```
[user@host ~]$ telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
* OK [CAPABILITY IMAP4REV1 LITERAL+ SASL-IR LOGIN-REFERRALS STARTTLS] localhost
  IMAP4rev1 2007a.403 at Sat, 14 Jun 2008 13:16:47 +0200 (CEST)

a001 CAPABILITY
* CAPABILITY IMAP4REV1 LITERAL+ IDLE UIDPLUS NAMESPACE CHILDREN MAILBOX-REFERRALS
  BINARY UNSELECT ESEARCH WITHIN SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT
  MULTIAPPEND SASL-IR LOGIN-REFERRALS STARTTLS
a001 OK CAPABILITY completed

a002 NOOP
a002 OK NOOP completed

a003 LOGIN pere pere
a003 OK [CAPABILITY IMAP4REV1 LITERAL+ IDLE UIDPLUS NAMESPACE CHILDREN MAILBOX-
  REFERRALS BINARY UNSELECT ESEARCH WITHIN SCAN SORT THREAD=REFERENCES
  THREAD=ORDEREDSUBJECT MULTIAPPEND] User pere authenticated

a004 SELECT inbox
* 4 EXISTS
* NO Mailbox vulnerable - directory /var/spool/mail must have 1777 protection
* 4 RECENT
* OK [UIDVALIDITY 1213385060] UID validity status
* OK [UIDNEXT 6] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/pere
* NO Mailbox vulnerable - directory /var/spool/mail must have 1777 protection
a004 OK [READ-WRITE] SELECT completed
```

```

a005 FETCH 1 all
* 1 FETCH (FLAGS (\Recent) INTERNALDATE "14-Jun-2008 13:14:49 +0200" RFC822.SIZE 627
  ENVELOPE ("Sat, 14 Jun 2008 13:14:47 +0200" "primer missatge" (...
a005 OK FETCH completed

a006 FETCH 1 full
* 1 FETCH (FLAGS (\Recent) INTERNALDATE "14-Jun-2008 13:14:49 +0200" RFC822.SIZE 627
  ENVELOPE ("Sat, 14 Jun 2008 13:14:47 +0200" "primer missatge" (...
a006 OK FETCH completed

a007 FETCH 1 body[header]
* 1 FETCH (BODY[HEADER] {547}
Return-Path: <root@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
  by tftp.server.cat (8.14.1/8.14.1) with SMTP id m5EBElas003535
  for <pere@tftp.server.cat>; Sat, 14 Jun 2008 13:14:49 +0200
Received: (from root@localhost)
  by tftp.server.cat (8.14.1/8.14.1/Submit) id m5EBElqf003534
  for pere; Sat, 14 Jun 2008 13:14:47 +0200
Date: Sat, 14 Jun 2008 13:14:47 +0200
From: root <root@tftp.server.cat>
Message-Id: <200806141114.m5EBElqf003534@tftp.server.cat>
To: pere@tftp.server.cat
Subject: primer missatge
)
* 1 FETCH (FLAGS (\Recent \Seen))
a007 OK FETCH completed

a008 FETCH 1 body
* 1 FETCH (BODY ("TEXT" "PLAIN" ("CHARSET" "X-UNKNOWN") NIL NIL "7BIT" 80 3))
a008 OK FETCH completed

a009 STORE 1 +flags \deleted
* 1 FETCH (FLAGS (\Recent \Seen \Deleted))
a009 OK STORE completed

a010 EXPUNGE
* 1 EXPUNGE
* 3 EXISTS
* 3 RECENT
* NO Mailbox vulnerable - directory /var/spool/mail must have 1777 protection
a010 OK Expunged 1 messages

a011 SELECT inbox
* 3 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1213385060] UID validity status
* OK [UIDNEXT 6] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/pere
* NO Mailbox vulnerable - directory /var/spool/mail must have 1777 protection
a011 OK [READ-WRITE] SELECT completed

a012 LOGOUT
* BYE portatil IMAP4rev1 server terminating connection
a012 OK LOGOUT completed
Connection closed by foreign host.
You have new mail in /var/spool/mail/root

```

Si fem el seguiment d'aquest diàleg podem observar que (en el llistat s'ha deixat una línia en blanc abans de cada ordre del client):

- Primerament s'entra en l'estat no autènticat i es realitzen les ordres *NOOP* i *CAPABILITY* per observar el mecanisme d'etiquetat de les comandes i les respostes.
- A continuació es valida l'usuari amb l'ordre *LOGIN* i s'entra en l'estat autènticat.
- Cal seleccionar una bústia per poder operar amb els seus missatges, això es fa amb la comanda *SELECT*. Si ens fixem en la resposta, la bústia inbox conté 4 missatges recents, el següent missatge que arribi tindrà el UID 6, i els flags són “\Answered \Flagged \Deleted \Draft \Seen”.
- Tot seguit es recupera el primer dels missatges amb l'ordre *FETCH* i es proven varies de les seves opcions.
- Amb l'ordre *STORE* es marca el missatge número 1 com a esborrant posant-li el flag “\Deleted”. L'ordre *EXPUNGE* l'elimina definitivament.
- Es pot observar que ara la bústia inbox conté únicament tres missatges.
- Per tancar una sessió IMAP s'utilitza la ordre *LOGOUT*.

Exemple 02: Sessió a un servidor local (2)

Aquesta és una altra sessió local a un servidor IMAP per l'accés remot al correu de l'usuari “pere”. Podeu obtenir una captura d'aquest diàleg feta amb Wireshark en el fitxer del material complementari:

- Complementari_AX_ud2_na3_imap_dialegLocal2.cap

!!

Es pot fer un seguiment i anàlisi de tot el diàleg d'aquest segon exemple consultant el fitxer de captura de Wireshark del material complementari.

```

[user@host ~]# telnet localhost 143
* OK [CAPABILITY IMAP4REV1 LITERAL+ SASL-IR LOGIN-REFERRALS STARTTLS] localhost
  IMAP4rev1 2007a.403 at Sat, 14 Jun 2008 13:43:25 +0200 (CEST)

a001 LOGIN pere pere
a001 OK [CAPABILITY IMAP4REV1 LITERAL+ IDLE UIDPLUS NAMESPACE CHILDREN MAILBOX-

```



```

REFERRALS BINARY UNSELECT ESEARCH WITHIN SCAN SORT THREAD=REFERENCES
THREAD=ORDEREDSUBJECT MULTIAPPEND] User pere authenticated

a002 SELECT inbox
* 3 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1213385060] UID validity status
* OK [UIDNEXT 6] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/pere
* NO Mailbox vulnerable - directory /var/spool/mail must have 1777 protection
a002 OK [READ-WRITE] SELECT completed

a003 FETCH 1 all
* 1 FETCH (FLAGS () INTERNALDATE "14-Jun-2008 13:15:30 +0200" RFC822.SIZE 602
  ENVELOPE ("Sat, 14 Jun 2008 13:15:28 +0200" "segon missatge" (...
a003 OK FETCH completed

a005 FETCH 1 envelope
* 1 FETCH (ENVELOPE "Sat, 14 Jun 2008 13:15:28 +0200" "segon missatge" (...
a005 OK FETCH completed

a006 FETCH 1 flags
* 1 FETCH (FLAGS ())
a006 OK FETCH completed

a007 FETCH 1 rfc822
* 1 FETCH (RFC822 {602}
Return-Path: <anna@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
  .by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5EBFSPm003595
  .for <pere@tftp.server.cat>; Sat, 14 Jun 2008 13:15:29 +0200
Received: (from anna@localhost)
  .by tftp.server.cat (8.14.1/8.14.1/Submit) id m5EBFS2Q003594
  .for pere; Sat, 14 Jun 2008 13:15:28 +0200
Date: Sat, 14 Jun 2008 13:15:28 +0200
From: anna@tftp.server.cat
Message-Id: <200806141115.m5EBFS2Q003594@tftp.server.cat>
To: pere@tftp.server.cat
Subject: segon missatge

exemple de missatge de la usuaria anna
a l'usuari pere
adeu
  FLAGS (\Seen))
a007 OK FETCH completed

a008 FETCH 1 rfc822.text
* 1 FETCH (RFC822.TEXT {63}
exemple de missatge de la usuaria anna
a l'usuari pere
adeu
)
a008 OK FETCH completed

a009 FETCH 1 rfc822.header
* 1 FETCH (RFC822.HEADER {539}
Return-Path: <anna@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
  .by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5EBFSPm003595
  .for <pere@tftp.server.cat>; Sat, 14 Jun 2008 13:15:29 +0200
Received: (from anna@localhost)
  .by tftp.server.cat (8.14.1/8.14.1/Submit) id m5EBFS2Q003594
  .for pere; Sat, 14 Jun 2008 13:15:28 +0200
Date: Sat, 14 Jun 2008 13:15:28 +0200
From: anna@tftp.server.cat
Message-Id: <200806141115.m5EBFS2Q003594@tftp.server.cat>
To: pere@tftp.server.cat
Subject: segon missatge

)
a009 OK FETCH completed

a010 FETCH 1 rfc822.size
* 1 FETCH (RFC822.SIZE 602)
a010 OK FETCH completed

a011 STORE 1 +flags \seen
* 1 FETCH (FLAGS (\Seen))
a011 OK STORE completed

a012 FETCH 1 flags
* 1 FETCH (FLAGS (\Seen))
a012 OK FETCH completed

a013 LOGOUT
* BYE portatil IMAP4rev1 server terminating connection
a013 OK Mailbox vulnerable - directory /var/spool/mail must have 1777 protection

```

Si es segueix el diàleg efectuat es pot observar que:

- La comanda *FETCH* proporciona una gran varietat de funcionalitats per descarregar un missatge de correu o parts d'ell.
- L'ordre per descarregar completament un missatge en format RFC822 és la comanda *FETCH n^o-missatge rfc822*, on l'argument "rfc822" indica el format de la descàrrega.
- Si l'argument és *rfc822.header* es descarreguen només les capçaleres.
- Si l'argument és *rfc822.text* es descarrega només el cos del missatge.
- Si l'argument és *rfc822.size* s'indica la mida en bytes del missatge.
- Els indicadors (flags) d'un missatge es poden canviar amb l'ordre *STORE*. Al primer missatge se li ha activat l'indicador que diu que el

missatge ja ha estat llegit.

Exemple 03: Sessió a un servidor local (3)

!!

El següent exemple mostra el funcionament del mecanisme de gestió de carpetes en un servidor IMAP. Podeu obtenir una captura d'aquest diàleg feta amb Wireshark en el fitxer del material complementari:

Es pot fer un seguiment i anàlisi de tot el diàleg d'aquest tercer exemple consultant el fitxer de captura de Wireshark del material complementari.

- Complementari_AX_ud2_na3_imap_dialogLocal3.cap

```
[root@portatil ~]# telnet localhost 143
* OK [CAPABILITY IMAP4REV1 LITERAL+ SASL-IR LOGIN-REFERRALS STARTTLS] localhost
IMAP4rev1 2007a.403 at Sat, 14 Jun 2008 14:02:36 +0200 (CEST)

a001 LOGIN pere pere
a001 OK [CAPABILITY IMAP4REV1 LITERAL+ IDLE UIDPLUS NAMESPACE CHILDREN MAILBOX-
REFERRALS BINARY UNSELECT ESEARCH WITHIN SCAN SORT THREAD=REFERENCES
THREAD=ORDEREDSUBJECT MULTIAPPEND] User pere authenticated

a002 CREATE personal
a002 OK CREATE completed

a003 SUBSCRIBE personal
a003 OK SUBSCRIBE completed

a004 SELECT personal
* 0 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1213445022] UID validity status
* OK [UIDNEXT 1] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
a004 OK [READ-WRITE] SELECT completed

a005 CLOSE
a005 OK CLOSE completed

a006 RENAME personal mybox
a006 OK RENAME completed

a007 SELECT inbox
* 3 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1213385060] UID validity status
* OK [UIDNEXT 6] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 2] first unseen message in /var/spool/mail/pere
* NO Mailbox vulnerable - directory /var/spool/mail must have 1777 protection
a007 OK [READ-WRITE] SELECT completed

a008 COPY 1:3 mybox
a008 OK [COPYUID 1213445022 3:5 1:3] COPY completed

a009 SELECT mybox
* 3 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1213445022] UID validity status
* OK [UIDNEXT 4] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 2] first unseen message in /tmp/mybox
a009 OK [READ-WRITE] SELECT completed

a010 FETCH 1:2 flags
* 1 FETCH (FLAGS (\Seen))
* 2 FETCH (FLAGS ())
a010 OK FETCH completed

a011 STORE 1:2 flags \answered \seen \Deleted
* 1 FETCH (FLAGS (\Seen \Deleted \Answered))
* 2 FETCH (FLAGS (\Seen \Deleted \Answered))
a011 OK STORE completed

a012 FETCH 1:2 flags
* 1 FETCH (FLAGS (\Seen \Deleted \Answered))
* 2 FETCH (FLAGS (\Seen \Deleted \Answered))
a012 OK FETCH completed

a015 LOGOUT
* BYE portatil IMAP4rev1 server terminating connection
a015 OK LOGOUT completed
```

Si es segueix el diàleg anterior s'observa que:

- Es poden crear carpetes al servidor amb l'ordre *CREATE*. Les carpetes s'activen amb l'ordre *SUBSCRIBE*.
- Es pot canviar de nom una carpeta creada amb l'ordre *RENAME*.
- Els missatges es poden copiar d'una bustia a una altra amb l'ordre *COPY*.
- Es poden assignar flags al un conjunt de missatges amb l'ordre *STORE*. L'assignació pot ser absoluta amb l'argument *flags* o relativa amb l'argument *+flags*.

Exemple 04: Sessió a un servidor remot

Utilitzant la utilitat **imtest** es pot fer una sessió a un compte de webmail com per exemple *Gmail*. Cal tenir present que cal indicar l'usuari per a l'autenticació, l'autorització i indicar que es tracta d'una comunicació

segura que utilitza SSL.

La utilitat **imtest** i la utilitat **pop3test** provenen del paquet “cyrus-imapd-utils” que és un dels paquets d'utilitats del servidor POP i IMAP Cyrus (un altre de molt utilitzat).

```
# Instal·lar el servidor pop i imap Cyrys
[root@host ~]# rpm -qf /usr/bin/pop3test
cyrus-imapd-utils-2.3.9-7.fc7
[root@phost ~]# rpm -qf /usr/bin/imtest
cyrus-imapd-utils-2.3.9-7.fc7
```

Tot següent es mostra el contingut d'una sessió remota de descàrrega de correu via IMAP a un servidor webmail com per exemple el de *Gmail*. Podeu obtenir una captura d'aquest diàleg feta amb wireshark en el fitxer del material complementari:

!!

Es pot fer un seguiment i anàlisi de tot el diàleg d'aquest quart exemple consultant el fitxer de captura de *wireshark* del material complementari.

- Complementari_AX_ud2_na3_imap_dialogGmail.cap

```
[root@host ~]# imtest -u ecanetlprofe -a ecanetlprofe -s imap.gmail.com
verify error:num=20:unable to get local issuer certificate
verify error:num=27:certificate not trusted
verify error:num=21:unable to verify the first certificate
TLS connection established: TLSv1 with cipher RC4-MD5 (128/128 bits)
S: * OK Gimap ready for requests from 81.34.247.52 u14if5376920gvf.0
C: C01 CAPABILITY
S: * CAPABILITY IMAP4rev1 UNSELECT IDLE NAMESPACE QUOTA XLIST CHILDREN XYZZY
S: C01 OK Thats all she wrote! u14if5376920gvf.0
Please enter your password:
C: L01 LOGIN ecanetlprofe {8}
S: + Ready u14if5376920gvf.0
C: <omitted>
S: L01 OK ecanetlprofe@gmail.com authenticated (Success)
Authenticated.
Security strength factor: 128

a001 CAPABILITY
* CAPABILITY IMAP4rev1 UNSELECT LITERAL+ IDLE NAMESPACE QUOTA ID XLIST CHILDREN
a001 OK Success

a002 SELECT inbox
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen *)]
* OK [UIDVALIDITY 2]
* 2 EXISTS
* 0 RECENT
* OK [UNSEEN 1]
* OK [UIDNEXT 49]
a002 OK [READ-WRITE] inbox selected. (Success)

a003 FETCH 1 rfc822.text
* 1 FETCH (RFC822.TEXT {16921}
-----=_Part_476_8183242.1213374666169
Content-Type: multipart/alternative;
        boundary="-----=_Part_477_4440518.1213374666170"
... output suprimit ...
a003 OK Success

a004 FETCH 1:2 rfc822.size
* 1 FETCH (RFC822.SIZE 17169)
* 2 FETCH (RFC822.SIZE 1402)
a004 OK Success

a005 FETCH 1:2 rfc822.header
* 1 FETCH (RFC822.HEADER {458}
Received: by 10.82.154.13 with HTTP; Fri, 13 Jun 2008 09:31:06 -0700 (PDT)
Message-ID: <34af42060806130931n46458220rae640f34f5e4bfb2@mail.gmail.com>
Date: Fri, 13 Jun 2008 18:31:06 +0200
From: "eduard canet" <ecanetlprofe@gmail.com>
To: ecanetlprofe@gmail.com
Subject: missatge de webmail amb contingut adjunt
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="-----=_Part_476_8183242.1213374666169"
Delivered-To: ecanetlprofe@gmail.com
)
* 2 FETCH (RFC822.HEADER {1328}
Delivered-To: ecanetlprofe@gmail.com
Received: by 10.82.154.13 with SMTP id b13cs8819bue; Fri, 13 Jun 2008 10:24:58
-0700 (PDT)
Received: by 10.210.41.14 with SMTP id o14mr2783168ebo.156.1213377897971; Fri,
13 Jun 2008 10:24:57 -0700 (PDT)
Return-Path: <anna@tftp.server.cat>
Received: from tftp.server.cat (52.Red-81-34-247.dynamicIP.rima-tde.net
[81.34.247.52]) by mx.google.com with ESMTP id
g1lsi5105113gve.8.2008.06.13.10.24.56; Fri, 13 Jun 2008 10:24:57 -0700 (PDT)
Received-SPF: error (google.com: error in processing during lookup of
anna@tftp.server.cat: DNS timeout) client-ip=81.34.247.52;
Authentication-Results: mx.google.com; spf=tempperror (google.com: error in
processing during lookup of anna@tftp.server.cat: DNS timeout)
smtp.mail=anna@tftp.server.cat
Received: from tftp.server.cat (localhost [127.0.0.1]) by tftp.server.cat
(8.14.1/8.14.1) with ESMTP id m5DH0e49005470 for <ecanetlprofe@gmail.com>;
Fri, 13 Jun 2008 19:24:41 +0200
Received: (from anna@localhost) by tftp.server.cat (8.14.1/8.14.1/Submit) id
m5DH0eUe005469 for ecanetlprofe@gmail.com; Fri, 13 Jun 2008 19:24:40 +0200
Date: Fri, 13 Jun 2008 19:24:40 +0200
From: anna@tftp.server.cat
Message-Id: <200806131724.m5DH0eUe005469@tftp.server.cat>
To: ecanetlprofe@gmail.com
Subject: missatge a l'exterior de anna
```

```

)
a005 OK Success
a006 LOGOUT
* BYE LOGOUT Requested
a006 OK 73 good day (Success)
Connection closed.

```

Si es segueix el contingut d'aquesta connexió IMAP sobre SSL es pot observar que:

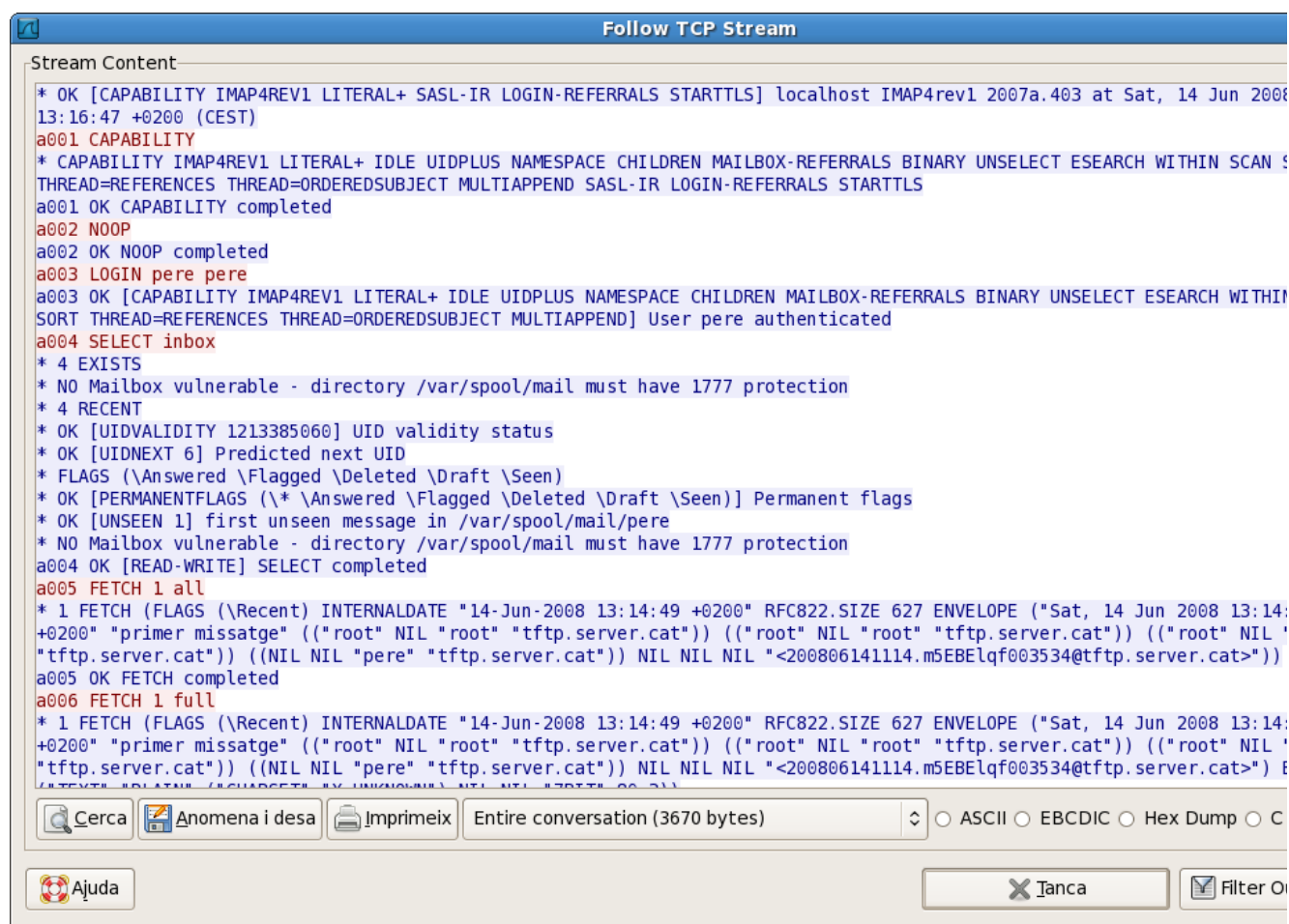
- L'usuari es valida contra el servidor de Gmail.
- Selecciona la carpeta *inbox* i descarrega tot el contingut del primer missatge en format *rfc822*. Com que el missatge conté un pdf és un missatge MIME multipart (s'ha suprimit de l'output tota la codificació en base64 del pdf).
- Es llista la mida en bytes dels dos primers missatges.
- Es descarreguen les capçaleres dels dos primers missatges.
- Finalment el client tanca la sessió.

Monitoritzar el tràfic IMAP

Si s'utilitza Wireshark per monitoritzar el tràfic d'una connexió IMAP es podran observar els ports client (dinàmics) i del servidor (143), el tipus de tràfic TCP, la connexió TCP de tres vies, les peticions del client i les respostes del servidor. En la Figura 13 "Pantalla de captura de wireshark d'un diàleg local IMAP" podeu observar en format text tot el diàleg IMAP entre un client i un servidor. Podeu obtenir una captura del wireshark del fitxer del material complementari:

- Complementari_AX_ud2_na3_imap_dialegLocal1.cap

Figura 13. Pantalla de captura de wireshark d'un diàleg local IMAP.



Un altre exemple de tràfic IMAP es pot realitzar entre un client de correu local com el *thunderbird* i un webmail com per exemple *Gmail*. La conversa amb el servidor IMAP de Gmail utilitzant SSL es pot consultar també amb el fitxer de material complementari:

- Complementari_AX_ud2_na3_imap_dialegGmail.cap

Existeixen multitud de clients de correu gràfics. Un dels més coneguts i utilitzats en GNU/Linux és el *thunderbird*, per a Windows un exemple de client de correu gràfic és el *Outlook*. Per configurar un compte de correu o més en un MUA com el *thunderbird* cal disposar d'un compte de correu en un servidor de correu. Els elements més importants en la configuració són:

- Identificar clarament el compte de correu de l'usuari i la identitat associada.
- Indicar quin és el servidor de correu sortint. El correu sortint generalment és enviat per un MTA que utilitza el protocol SMTP. Sovint els administradors dels dominis han posat alias al servidor de correu de manera que s'identifica al servidor sortint amb noms del tipus *smtp.servidor.domini*.
- Per tal de recuperar el correu del servidor cal indicar el nom del servidor i el protocol a utilitzar per l'accés remot a la bústia de correu. El protocol usat generalment és POP o IMAP. Molts ISP han posat alias als servidors de manera que les màquines tenen noms del tipus *pop.servidor.domini* o *imap.servidor.domini*. Per accedir al correu cal identificar-se indicant el nom de l'usuari i sovint cal indicar una contrasenya.
- En entorns de comunicació que exigeixen privacitat de les dades cal configurar correctament el mecanisme de transport segur a utilitzar, SSL o TLS i configurar-lo correctament.

Sempre que es configura un compte de correu en un client de correu cal disposar de la següent informació:

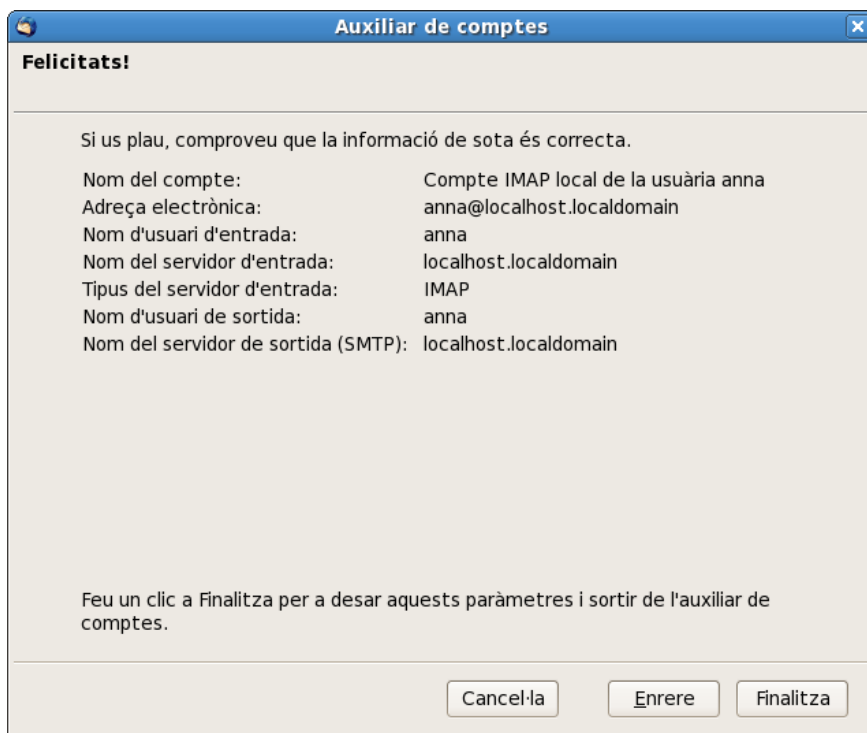
- correu **SMTP sortint**: nom del servidor, port i aspectes de seguretat i autenticació.
- correu **entrant (POP o IMAP)**: nom del servidor, port i aspectes de seguretat i autenticació.

Si es vol crear un compte de correu al *thunderbird* que permeti accedir al correu de l'usuària "anna" a través del servidor IMAP local, un repàs ràpid als passos a efectuar són:

- seleccionar que es vol crear un nou compte de correu.
- indicar el nom de l'usuari ("anna") i el seu correu (anna@localhost.localdomain).
- indicar el nom del servidor de correu SMTP sortint (localhost.localdomain perquè s'utilitza el *sendmail* local) i el del servidor de correu IMAP entrant (localhost.localdomain perquè s'utilitza el servidor IMAP local).
- indicar el nom amb el que s'identificarà aquest compte dins del thunderbird ("anna"). Si un usuari configura més d'un compte de correu al *thunderbird* convé que assigni a cada compte noms clarificadors, per exemple: "anna personal", "anna feina", "anna a gmail", etc.
- escriure una descripció del compte (compte de prova per a l'usuària local anna").

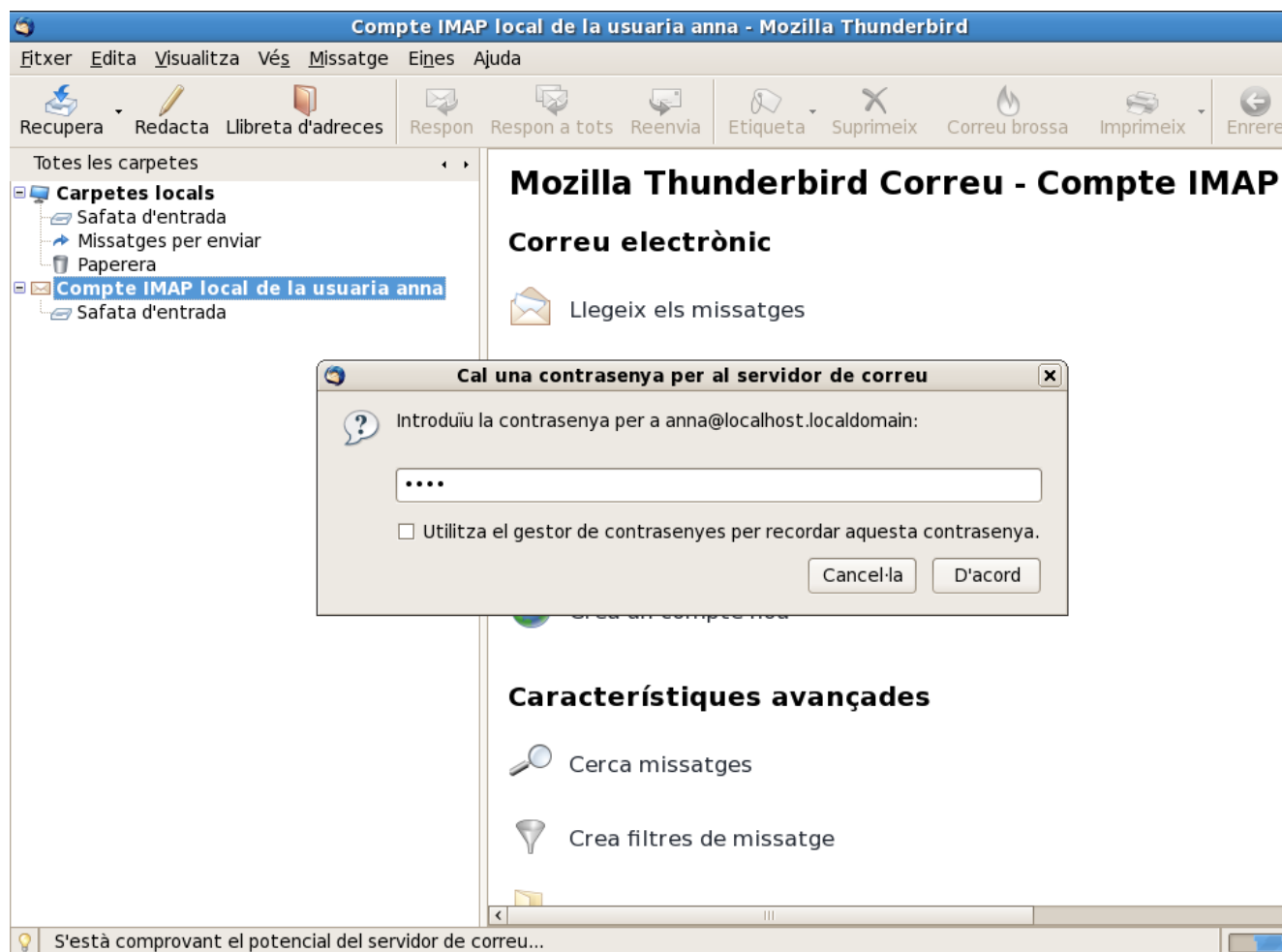
La Figura 14 "Panell resum de la creació d'un compte de correu local en el thunderbird usant IMAP" mostra la pantalla resum del *thunderbird* que es mostra en finalitzar la creació d'un compte de correu. En aquesta pantalla d'indiquen totes aquelles dades necessàries per configurar apropiadament l'accés al servidor.

Figura 14. Panell resum de la creació d'un compte de correu local en el thunderbird usant IMAP.



La Figura 15 "Pantalla del compte d'usuari IMAP en el thunderbird" mostra el compte d'un usuari local en el *thunderbird*, que accedeix via IMAP al servidor local. Es pot observar que es requereix validar el compte d'usuari introduïnt el password.

Figura 15. Pantalla del compte d'usuari IMAP en el thunderbird.



Client gràfic thunderbird: Servidor IMAP webmail

La majoria de servidors Webmail com *Gmail*, *Yahoo*, etc permeten descarregar-se el correu des d'un client POP o IMAP. Podem configurar Thunderbird per connectar amb el servidor de Gmail.

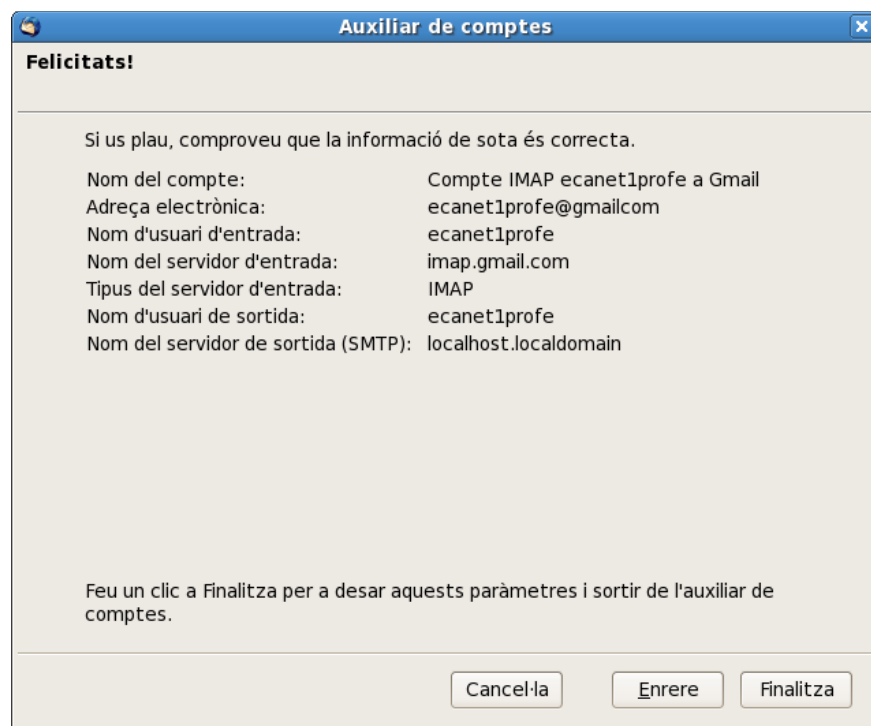
Primerament cal configurar *Gmail* per permetre fer això, cal anar a *settings* i a *forwarding POP & IMAP*. Cal assegurar-se de deixar activat que es permet la descàrrega per POP i/o per IMAP. Gmail proporciona instruccions de configuració de clients gràfics com thunderbird (entre altres) per descarregar el correu de Gmail.

Un cop fet això es pot configurar el compte d'usuari de Gmail en el Thunderbird per descarregar-lo via IMAP:

- Indicar el nom propi de l'usuari i el compte de correu a usar (nom del compte complert, usuari @domini).
- Indicar el tipus d'accés remot que s'utilitzarà POP o IMAP i el nom del servidor a utilitzar.
- Indicar el nom d'usuari del compte de correu.
- Escriure un identificador del compte de correu. Si voleu es pot deixar el nom del propi compte, si en teniu molts potser es millor escriure una frase descriptiva del tipus: compte de la feina, compte de casa, compte dels amics, compte per a delictes, etc.
- Un cop indicades totes les dades es mostra el resum.
- Un cop creat el compte apareix la pantalla de treball de Thunderbird. Encara cal configurar la seguretat.
- La comunicació amb Gmail ha de ser xifrada, per això cal activar SSL.
- En la configuració del servidor de correu sortint cal indicar que s'utilitzi TLS si disposa de correu segur.

La Figura 16 "Pantalla de resum de descàrrega IMAP de Gmail" mostra la pantalla de configuració resum del *thunderbird* per descarregar correu via IMAP de Gmail. Aquesta és la informació necessària per configurar apropiadament l'accés via IMAP a un compte de *Gmail*.

Figura 16. Pantalla de resum de descàrrega IMAP de Gmail.



1.6. Usa clients de correu electrònic per enviar i rebre correu des dels comptes creats en el servidor.

Tal i com s'ha pogut apreciar en els diversos exemples d'aquesta documentació existeixen molts clients de correu de característiques molt

diverses. En general es poden classificar en:

- Clients de text: són aquelles utilitats Unix i GNU/Linux de tota la vida que permeten generar missatges de correu i accedir a la pròpia bústia de correu. Són utilitats de consola, és a dir, de text. N'hi ha de format ben simple, com l'ordre *mail* i evolucions que permeten treballar amb programes de menús en color també en entorn de consola com per exemple *mutt*.
- Clients gràfics: amb la popularització dels entorns gràfics van sorgir programes client de correu com *eudora*, *evince*, *Outlook*, *thunderbird*, etc. Durant un període de temps aquests programes eren l'eina més usada pels usuaris per accedir a les seves bústies de correu.
- Client webmail: actualment la majoria de comptes de correus dels usuaris són basats en webmail, principalment en serveis de webmail gratuïts com *Gmail*, *Yahoo*, *Hotmail*, etc. Fins i tot els enorns corporatius utilitzen correu webmail hostatjat en serveis externalitats.

1.6.1. Client de text: email

Una de les ordres més utilitzades per a la gestió de correu en entorns Unix i GNU/Linux és la simple ordre **mail**. Amb aquesta ordre es poden redactar i recuperar correus de text molt fàcilment. Cada usuari del sistema disposa d'un compte de correu accessible via mail per defecte.

Les principals opcions que permet l'ordre mail són:

- enviar correus de text pla.
- enviar correus amb 'attachments'.
- desar en fitxers externs el contingut del correu (tant el text pla com qualsevol altre tipus de contingut).
- llistar els correus de la bústia.
- gestionar els missatges de la bústia: eliminar, seleccionar, reenviar.

El següent llistat de codi mostra com l'usuari pere s'envia emails a ell mateix i posteriorment consulta la seva bústia de missatges.

```
#Redactar un email per a ell mateix:
[pere@host ~]$mail -s nou "email per en pere" pere
hola
aquest és un nou email per a l'usuari pere
s'ha escrit usant l'ordre email
el seu contingut anirà al mailbox
de l'usuari pere
adeu
.
Cc:

#Accedir als missatges:
[pere@host ~]$mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/pere": 2 messages 2 new
>N 1 pere@pc84.informatic Fri Jun 13 12:32 22/924 "nou"
N 2 pere@pc84.informatic Fri Jun 13 12:34 21/916 "nou"
&l
Message 1:
From pere@pc84.informatica.escoladeltreball.org Fri Jun 13 12:32:59 2008
Date: Fri, 13 Jun 2008 12:32:59 +0200
From: pere@pc84.informatica.escoladeltreball.org
To: email.per.en.pere@pc84.informatica.escoladeltreball.org,
pere@pc84.informatica.escoladeltreball.org
Subject: nou

hola
aquest és un nou email per a l'usuari pere
s'ha escrit usant l'ordre email
el seu contingut anirà al mailbox
de l'usuari pere
adeu

& q
Saved 1 message in mbox
Held 1 message in /var/spool/mail/pere
You have mail in /var/spool/mail/pere
```

L'ordre *mail* també permet enviar fitxers adjunts, tot i que d'una manera molt simplificada. Per adjuntar el fitxer s'utilitza el redireccionament d'entrada "<". Un cop rebut el missatge (que de fet conté el fitxer) el destinatari el pot desar a part, en un fitxer, usant l'opció *write*.

```
# En pere envia un pdf a l'anna
```



```
[pere@host ~]$ mail -s "Dossier m08" anna < dossier.pdf
# L'anna rep el missatge i desa el pdf adjunt en un fitxer
[anna@host ~]$ mail
Heirloom Mail version 12.5 7/5/10. Type ? for help.
"/var/spool/mail/anna": 5 messages 1 new
  1 anna@localhost6.local Sat Jan 28 18:16 56/2706 "ubj"
...
>N 5 pere@localhost6.local Wed Feb 1 14:10 34/1581 "Dos"
& write 5 dossier.pdf
"dossier.pdf" [New file] 15/791
```

El següent llistat de codi mostra com adjuntar més d'un fitxer agrupant-los en un *.tar*. De fet es tracta d'un *.tgz* perquè es comprimeix. L'usuari "pere" envia al'usuaria "anna" el contingut d'un directori agrupat en un *tgz*.

```
# Generar un tar d'un directori i enviar-lo a anna
[pere@host ~]$ tar cvzf dir.tgz *
[pere@host ~]$ mail -s "tar del directori" anna < dir.tgz

# Anna desa el tar adjunt en un fitxer independent.
[anna@host ~]$ mail
Heirloom Mail version 12.5 7/5/10. Type ? for help.
"/var/spool/mail/anna": 7 messages 1 unread
...
>U 7 pere@localhost6.local Wed Feb 1 14:22 363/25835 "tar del directori"
& w 7 dir.tgz
"dir.tgz" [New file] 0/18515

[anna@host ~]$ ll dir.tgz
-rw-rw-r-- 1 anna anna 18515 1 feb 14:23 dir.tgz
```

De fet el contingut delsemails que s'envia com a fitxer adjunt no cal que provingui d'un fitxer, pot precedir de qualsevol contingut generat per l'entrada estàndard. El següent exemple envia en un paquet *.tgz* tots els fitxers pdf del directori actiu sense necessitat de generar el fitxer *tarball*.

```
# Envia adjunt un tar amb els pdf, sense crear el fitxer tar
[pere@host ~]$ tar cvz *.pdf | mail -s "tar on-the-fly" anna
dossier.pdf
treball.pdf

# Desar el contingut com un .tgz
[anna@host ~]$ mail
Heirloom Mail version 12.5 7/5/10. Type ? for help.
"/var/spool/mail/anna": 7 messages 1 unread
...
>N 8 pere@localhost6.local Wed Feb 1 14:27 20/890 "tar on-the-fly"
& w 8 pdfs.tgz
"pdfs.tgz" [New file] 0/18515

# Observar que conté els mateixos pdf
[anna@portatil ~]$ tar tvzf pdfs.tgz
-rw-rw-r-- pere/pere 791 2012-02-01 14:10 dossier.pdf
-rw-rw-r-- pere/pere 895 2012-02-01 14:27 treball.pdf
```

Es pot aprendre més del funcionament de l'ordre consultant les seves pàgines de manual amb l'ordre **man mail**, o interactivament dins d'una sessió de mail executar l'opció *help*.

```
[anna@host ~]$ mail
& help
mail commands
type <message list>      type messages
next                     goto and type next message
from <message list>      give head lines of messages
headers                  print out active message headers
delete <message list>    delete messages
undelete <message list>  undelete messages
save <message list> folder append messages to folder and mark as saved
copy <message list> folder append messages to folder without marking them
write <message list> file  append message texts to file, save attachments
preserve <message list>  keep incoming messages in mailbox even if saved
Reply <message list>      reply to message senders
reply <message list>      reply to message senders and all recipients
mail addresses           mail to specific recipients
file folder              change to another folder
quit                     quit and apply changes to folder
xit                      quit and discard changes made to folder
!                         shell escape
cd <directory>           chdir to directory or home if none given
list                     list names of all available commands
A <message list> consists of integers, ranges of same, or other criteria
separated by spaces. If omitted, mail uses the last message typed.
```

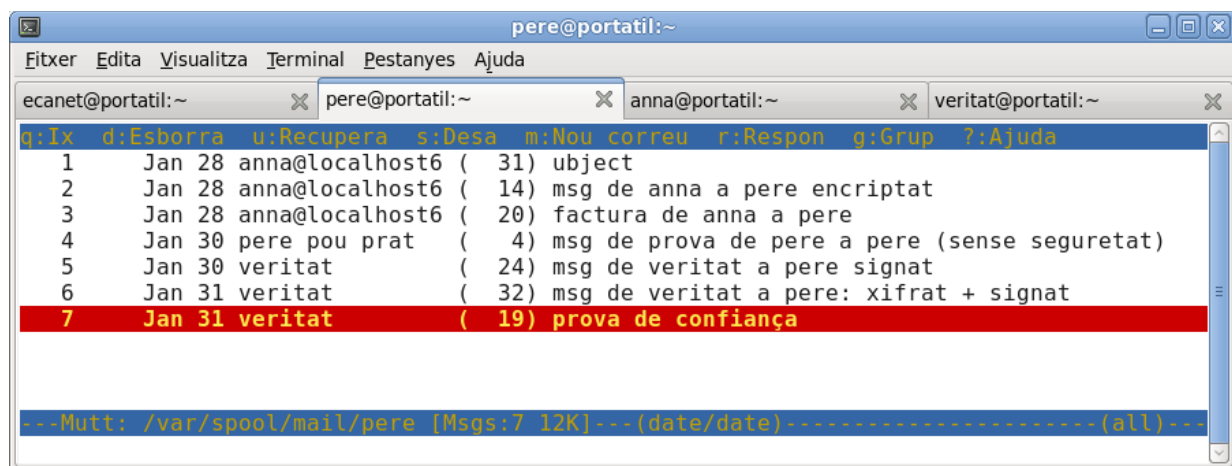
1.6.2.Client de text: mutt

A part de la clàssica ordre *mail* existeixen altres clients de text que faciliten la gestió de fitxers adjunts (una mica engorrosa amb *mail*). Un dels clients MUA en mode text és el *mutt*, que proporciona una interfície de menús per a la gestió de la bústia de correu.

La Figura 17 "Pantalla de treball de *mutt*" permet observar la gestió de correu que fa *mutt* a través de consola en forma text. L'usuari es pot desplaçar per els missatges i escollir quin vol seleccionar. Per redactar missatges *mutt* crida a l'editor per defecte que en aquest cas és l'aplicació *vim*. Com es pot observar de la barra d'opcions superior es permet:

- (d) esborrar el missatge seleccionat.
- (u) recuperar un missatge esborrat. Anular l'acció anterior.
- (s) desar a disc un missatge i els seus adjunts.
- (m) redactar un missatge nou.
- (r) respondre a un missatge existent.
- (g) group.
- (?) mostrar l'ajuda.
- (n) cerca de text en els missatges.
- (a) adjuntar fitxers (en la fase de redacció d'un missatge).
- (v) desar (s) els fitxers adjunts (en la fase de visualització d'un missatge rebut).

Figura 17. Pantalla de treball de *mutt*.



El procés a seguir per a la utilització de *mutt* és el següent:

- Un cop instal·lat el programari executar-lo des d'un compte d'usuari.
- L'aplicació demana si es vol crear o no la bústia de correu de l'usuari pròpia del *mutt*. Aquesta bústia es desa en el directori `/home/<usuari>/Mail`.
- S'accedeix a la bústia de missatges i es mostren. L'usuari pot redactar missatges nous, adjuntar fitxers i gestionar els missatges existents.

1.6.3. Clients gràfics: thunderbird

Existeixen multitud de clients de correu gràfics. Un dels més coneguts i utilitzats en GNU/Linux és el *thunderbird*, per a Windows un exemple de client de correu gràfic és el *Outlook*.

Per configurar un compte de correu o més en un MUA com el *thunderbird* cal disposar d'un compte de correu en un servidor de correu. Aquest compte pot estar en qualsevol servidor que accepti l'accés remot usant POP o bé IMAP. Es poden crear comptes de correu que permetin l'accés a bústies d'un servidor local (com els instal·lats en altres apartats) o que accedeixin a comptes de servidors externs, com per exemple l'accés al correu de *Gmail*. Els elements més importants en la configuració són:

- Identificar clarament el compte de correu de l'usuari i la identitat associada.

- Indicar quin és el servidor de correu sortint. El correu sortint generalment és enviat per un MTA que utilitza el protocol SMTP. Sovint els administradors dels dominis han posat alias al servidor de correu de manera que s'identifica al servidor sortint amb noms del tipus *smtp.servidor.domini*.
- Per tal de recuperar el correu del servidor cal indicar el nom del servidor i el protocol a utilitzar per l'accés remot a la bústia de correu. El protocol usat generalment és POP o IMAP. Molts ISP han posat àlies als servidors de manera que les màquines tenen noms del tipus *pop.servidor.domini* o *imap.servidor.domini*. Per accedir al correu cal identificar-se indicant el nom de l'usuari i sovint cal indicar una contrassenya.
- En entorns de comunicació que exigeixen privacitat de les dades cal configurar correctament el mecanisme de transport segur a utilitzar, SSL o TLS i configurar-lo correctament.

La Figura 18 “Parametres d'un compte en el thunderbird” mostra els paràmetres de configuració del *thunderbird* per a l'usuari pere. Aquest panell és el principal punt d'administració dels comptes de correu i l'accés als servidors.

Figura 18. Parametres d'un compte en el thunderbird.

The screenshot shows the 'Paràmetres del compte' (Account Settings) window in Thunderbird. The left sidebar lists various settings categories for the selected account, including server parameters, copies and folders, redaction and addressing, mail news parameters, OpenPGP security, reception confirmations, and local folders. The main pane is titled 'Paràmetres del compte - <pere@localhost.localdomain>' and contains the following fields and options:

- Nom del compte:** A text field containing 'pere@localhost.localdomain'.
- Identitat:** A section with a description: 'Cada compte pot tenir una identitat pròpia, que és la informació que els altres veuen quan llegeixen els vostres missatges.'
 - El vostre nom:** A text field containing 'pere pou prat'.
 - Adreça electrònica:** A text field containing 'pere@localhost.localdomain'.
 - Adreça de resposta:** An empty text field.
 - Organització:** An empty text field.
 - Text de la signatura:** A checkbox labeled 'Utilitza HTML (ex., bold)' is unchecked. Below it is a large empty text area for the signature.
 - Adjunta la signatura des d'un fitxer (text, HTML, o imatges):** An unchecked checkbox. Below it is a text field and a 'Trieu...' button.
 - Adjunta la meua targeta electrònica als missatges (vCard):** An unchecked checkbox. To its right is a button labeled 'Edita la targeta electrònica...'.
 - Servidor de sortida (SMTP):** A dropdown menu showing 'pere - localhost.localdomain (Per defecte)'.
 - Gestiona les identitats...** A button.
- Accions del compte:** A dropdown menu at the bottom left.
- Buttons:** At the bottom right, there are 'Cancel·la' (with a red X icon) and 'D'acord' (with a blue arrow icon) buttons.

Conceptes clau

Observant l'informació que mostra el panell de “paràmetres del compte” s'identifiquen els següents conceptes claus en la gestió de l'aplicació:

- Identitats (els diversos comptes de correu d'un usuari).
- Carpetes.
- Servidor de sortida SMTP.
- Paràmetres del servidor [d'entrada POP o IMAP] (configurat per a cada compte).

Identitats

Un usuari pot disposar de varis comptes de correu que s'anomenen *identitats*. En l'exemple l'usuari “pere” disposa d'un compte de correu en el servidor local i d'un compte de correu extern a gmail amb l'usuari “ecanet1profe”. Es poden crear tantes identitats com facin falta. Així si un usuari té cinc comptes de correu repartits en diferents servidors (un local, dos a *Gmail* i dos a *Yahoo*) pot crear les identitats per accedir als cinc comptes de correu i centralitzar-ho tot a través del *thunderbird*. Es pot observar que per a cada identitat hi ha el mateix llistat d'opcions de configuració del compte.

Per identificar fàcilment les identitats és aconsellable posar una descripció i/o nom que permeti entendre fàcilment la seva finalitat. Per exemple “pere feina”, “pere amics”, “pere jocs” i “soltitari-lliure” permeten tenir clar a que dedica l'usuari “pere” cada un dels comptes (l'últim...). L'identitat del compte local que es mostra és “pere pou prat”, la identitat del compte de gmail és “professor”.

Els paràmetres que descriuen una identitat són:

- *nom del compte*: identifica amb un nom el compte de correu. Pot ser un nom explicatiu.
- *nom*: aquest és el nom que rebran els destinataris en rebre els emails d'aquest compte.
- *adreça*: identifica l'adreça de correu a usar.
- *resposta*: identifica l'adreça de correu on s'han de dirigir les respostes. Si l'usuari té múltiples comptes pot interessar-li agrupar totes les respostes en un únic compte. Per exemple totes les respostes van al departament de reclamacions.
- *organització*: identifica l'empresa o organització a la que pertany el compte de correu.
- *signatura*: quan s'envia un correu es pot afegir al final una signatura o peu del correu amb informació de l'usuari o de l'empresa. Un exemple és el típic missatge legal de no enviar spam o no llegir si el correu no és per tu. D'altres hi posen la seva informació personal de contacte o alguna cita famosa. Aquest contingut pot ser text,html o provenir d'un fitxer extern.
- *targeta de visita*: existeix un mecanisme estandaritzat de lliurar conjuntament amb el correu una targeta de visita de l'emissor, igual que fa un venedor quan lliura al client una targeta. Les targetes de visita són en format vCard.
- *Servidor de sortida*: per a cada compte es pot indicar quin dels servidors de sortida SMTP configurats es vol utilitzar. Es pot observar que l'usuari “pere” utilitza el servidor local i en canvi l'usuari de *Gmail* utilitza el seu servidor.

Carpetes

Existeix un compte especial que no està associat a cap compte local anomenat “carpetes locals”. Aquest compte permet la configuració de diversos aspectes de funcionament local del *thunderbird* com són la gestió del “correu brossa” i la gestió de “l'espai de disc”.

- *correu brossa*: permet definir les característiques que fan que un correu sigui identificat com a spam. Per exemple permet usar *spamAssassin* o *spamPal*. Un cop identificat un correu com a spam cal decidir què fer-ne, on dipositar-lo. Tots aquests aspectes són governats aquí.
- *espai de disc*: permet indicar la política d'eliminació de missatges per preservar espai de disc, indicant uns màxims de dies o d'espai.

Servidor de sortida SMTP

Per poder enviar missatges de correu des del *thunderbird* cal que aquest es connecti via SMTP amb un servidor MTA. En aquest panell es poden crear, editar i suprimir servidors SMTP i escollir quin d'ells és per defecte.

Els comptes d'usuari locals (com el de l'usuari “pere”) utilitzen el servidor SMTP local (instal·lat com a pràctica). Els comptes externs com els de *Gmail* han de contactar directament via SMTP amb el seu

servidor. Es poden definir tants servidors SMTP com facin falta.

La Figura 19 "Paràmetres SMTP del servidor local" mostra els valors necessaris per configurar una connexió SMTP amb el servidor local instal·lat per defecte.

Figura 19. Paràmetres SMTP del servidor local.



La configuració del "correu de sortida" és important perquè si no es fa bé no es podran enviar correus. Els camps que cal definir apropiadament són:

- **Descripció:** un text identificatiu del tipus de la connexió SMTP que s'està configurant. Per exemple "servidor seu de girona" o "servidor de gmail".
- **Nom del servidor:** nom de host del servidor (per exemple smtp.domini.cat).
- **Port:** típicament el port 25 per a connexions smtp en text pla i 465 per a connexions segures via SSL.
- **Utilitza nom i contrassenya:** Si s'activa el client *thunderbird* intenta identificar/validar l'usuari enviant el nom i la contrassenya en contactar amb el servidor SMTP. Això s'ha de seleccionar si el servidor SMTP requereix d'autenticació de l'usuari.
- **Nom usuari:** (només si s'ha activat usar nom i contrassenya) indica el nom de l'usuari en el servidor SMTP.
- **Autenticació segura:** (només si s'ha activat usar nom i contrassenya) Indica si l'autenticació ha de ser en format segur o no.
- **Connexió segura:** permet escollir si la connexió es fa en text pla o xifrada usant STARTTLS o SSL.

Paràmetres del servidor d'entrada

Per a cada compte de correu configurat (o identitat) cal establir els paràmetres de configuració del servidor d'entrada, d'on s'obté el correu. Aquest accés es realitza utilitzant algun dels protocols d'accés remot a la bústia de correu, protocols POP o IMAP. Si no es configuren apropiadament els valors del servidor serà impossible accedir a la bústia de correu remota per descarregar els missatges.

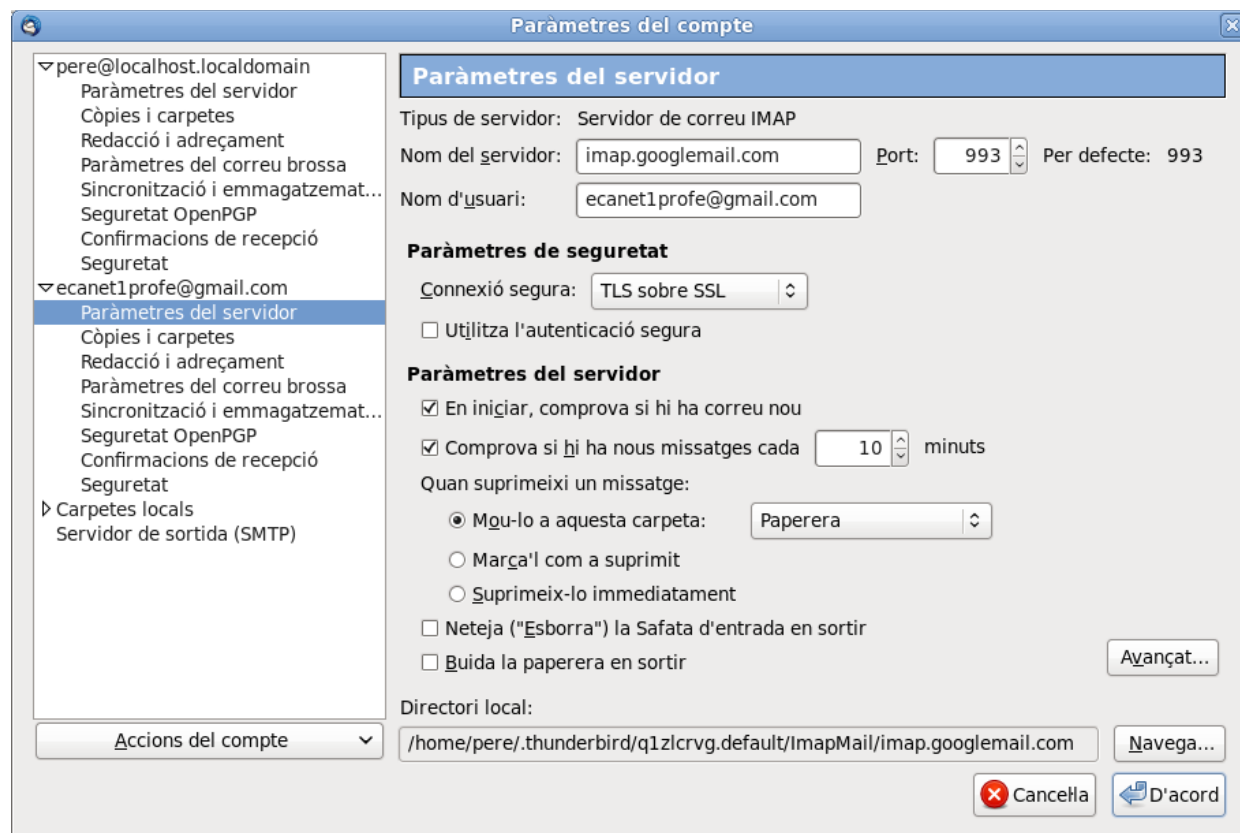
Les opcions de configuració que s'hi detallen són:

- **Tipus de servidor** especifica si s'accedeix a un servidor IMAP o a un servidor POP.
- **Nom del servidor:** indica el nom de host del servidor, del tipus imap.domini.com o pop.domini.cat.
- **Port:** especifica el port a usar. El protocol IMAP usa per defecte el port 143 per al transport en text pla i el 993 per al transport segur de IMAP usant SSL. El protocol POP utilitza el port 110 per a transmissions en text pla i el port 995 per a connexions segures.
- **Nom usuari:** indica el nom del compte de correu de l'usuari en aquell servidor.
- **Connexió segura:** si cal usar SSL/TLS o STARTTLS per xifrar la comunicació o treballar amb connexió usant text pla.
- **Autenticació segura:** indica si cal que el client s'autentifiqui l'usuari en iniciar la sessió en el servidor.
- **Paràmetres del servidor:** engloba un conjunt d'opcions que permeten establir cada quan comprovar si hi ha correu nou, què fer quan s'esborra un missatge, què fer en sortir i indicar la ubicació dels fitxers

'reals del sistema' on hi ha les carpetes personals del compte de correu.

La Figura 20 "Paràmetres del Servidor entrant IMAP" mostra la configuració de l'accés al servidor IMAP de *Gmail*. Aquest accés és usant una connexió segura SSL al port 993 del servidor IMAP anomenat *imap.googlemail.com*.

Figura 20. Paràmetres del servidor entrant IMAP.



Crear un compte d'usuari

El procés a seguir per crear comptes d'usuari en el client gràfic *thunderbird* és:

1. Accedir al panell "paràmetres del compte" i seleccionar dins "d'accions del compte" l'opció de "crear un compte nou".
2. El panell de configuració del compte requereix les dades del *nom*, *email* i *password* associats al compte. El password s'utilitza per validar que qui accedeix al correu thunderbird és qui realment ha configurat el compte.
3. Un cop s'ha indicat l'*email* a usar l'aplicació utilitza el domini indicat per contactar automàticament amb el servidor i obtenir automàticament els valors de "servidor de sortida SMTP" i "Servidor d'entrada POP o IMAP". En general els valors detectats seran els apropiats. Si no ho són o es vol modificar-ne alguna particularitat existeix un botó per poder fer la configuració manual d'aquests valors.

En el cas d'un compte al servidor SMTP local (configurat per defecte) cal tenir present que:

- Es pot mostrar una advertència o "Excepció de seguretat" indicant que s'accedeix a un servidor en text pla, sense usar SSL. Evidentment això genera tràfic insegur que pot ser monitoritzat per 'tercers' en la xarxa. Però si el servidor instal·lat no implementa SSL cal acceptar "comprenc els riscos".
- En el cas de la configuració SMTP local cal assegurar-se que NO s'ha seleccionat "Utilitza el nom i la contrassenya" en l'apartat de "Seguretat i autenticació". Així com no implementar "CAP" mecanisme de connexió segura (ni SSL ni STARTTLS).

En el cas d'un compte a un servidor externs com per exemple *Gmail* cal assegurar-se d'establir les opcions SMTP tal i com es mostren en la Figura 21 "Paràmetres SMTP del servidor Gmail". Es pot observar que es realitza una connexió segura amb TLS al port 465 del servidor *smtp.googlemail.com*.

Figura 21. Paràmetres SMTP del servidor Gmail.

Opcions de configuració

Un cop creat un compte de correu o "indentitat" es poden personalitzar tots els detalls que es desitgin a partir de les opcions del panell "Paràmetres del compte" tal i com es pot observar en la Figura 18 "Parametres d'un compte en el thunderbird". La configuració es classifica en els següents apartats:

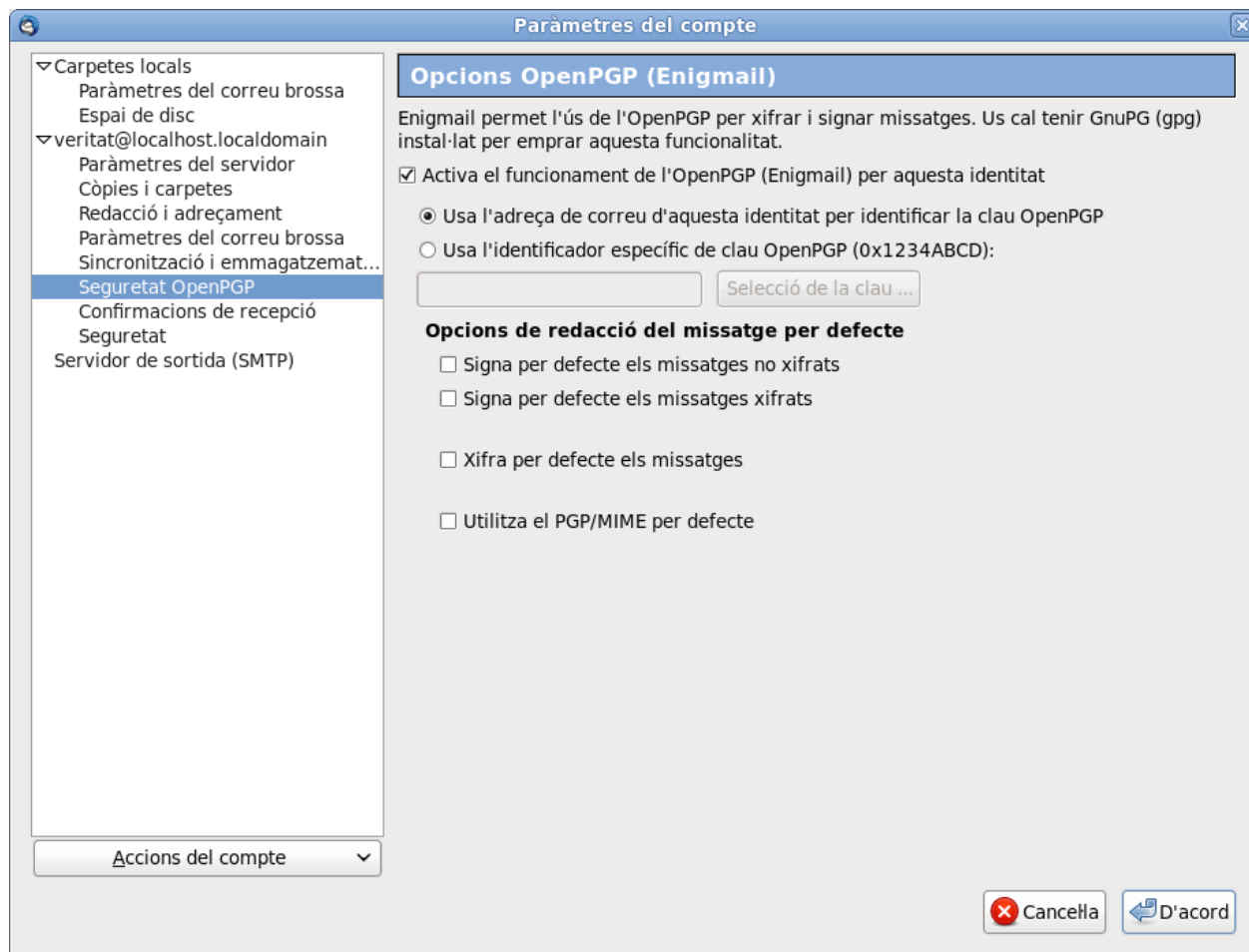
- *Paràmetres del servidor*: Cal especificar quin és el servidor de sortida a usar, port i usuari; les opcions de seguretat i les opcions de configuració generals (intervalls de connexió, comprovar correu nou, etc).
- *Còpies i carpetes*: gestiona les carpetes, esborranys, plantilles i les còpies dels missatges.
- *Redacció i adreçament*: Permet seleccionar si la redacció dels missatges és en format text o html, si cal incloure els missatges originals en les respostes, etc.
- *Paràmetres de correu brossa*: Estableix els paràmetres de gestió de l'spam.
- *Sincronització i emmagatzemament*: Estableix la sincronització amb el servidor i la gestió de l'espai d'emmagatzemament.
- *Seguretat Open PGP*: en aquest panell és on es determina la configuració de **OpenPGP** que permet la transmissió de missatges signats i xifrats.
- *Confirmació recepció*: permet especificar el mecanisme de confirmació dels missatges rebuts.
- *Seguretat*: En quest panell es realitza la configuració dels **Certificats Digitals** necessàris per a la transmissió de missatges xifrats i signats usant **S/MIME**.

1.6.4. Clients webmail

Avui en dia tothom utilitza almenys un compte de correu en algun dels nombrosos serveis *Webmail* que s'ofereixen. De fet sovint els usuaris disposen de múltiples comptes de correus en proveïdors diferents. És més, segurament disposen de més d'un compte (per a finalitats diferents) en un mateix servidor. En la Figura 22 "Servei web de Gmail" es pot veure la pantalla d'entrada al servei de correu web de *Google*. En aquesta pantalla els clients s'han d'identificar indicant l'usuari i la contrassenya.

Com es pot observar es tracta d'un servei gratuït en que els usuaris es poden donar d'alta sense pagar res (més anllà de la publicitat que han de suportar).

Figura 22. Servei web de Gmail.

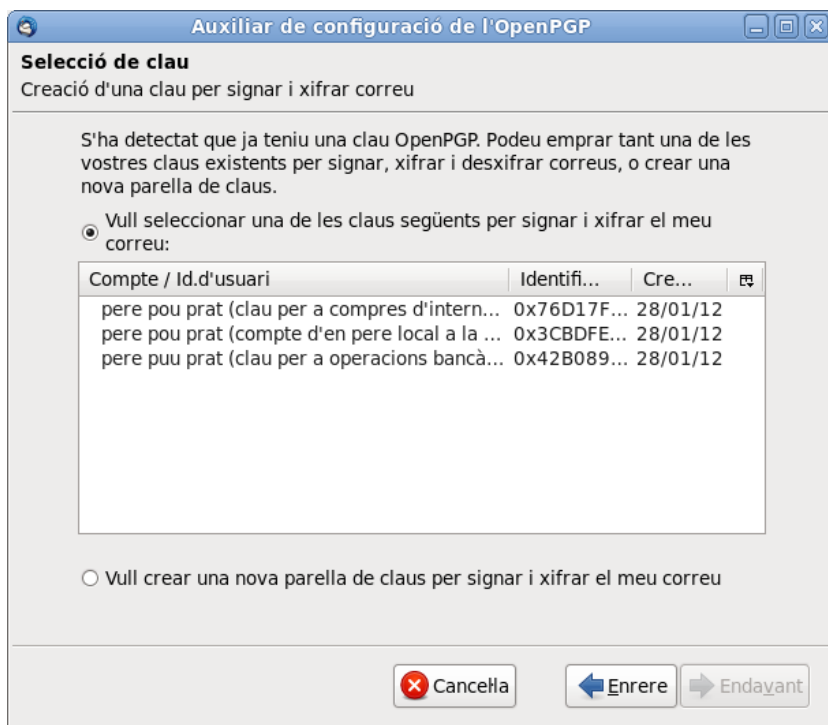


L'externalització de serveis i de servidors creix cada vegada més en el camp de l'informàtica. Una opció per als administradors informàtics d'una xarxa corporativa és "externalitzar" el servei de correu a un proveïdor extern. És a dir, en lloc de gestionar el correu des de la pròpia empresa es paga a un tercer perquè en porti aquesta gestió.

De fet *Gmail* proporciona de fa temps aquest servei però no gratuïtament. Els administradors locals de l'empresa han de gestionar-ne l'aspecte, la personalització i gestionar quins usuaris i llistes d'usuaris pertanyen al servei. Però tota la gestió del correu es fa amb les mateixes prestacions que amb la web de *Gmail*. És a dir, tens el teu correu de l'empresa però el tens dins de gmail, amb tota la potència de les seves eines.

En la Figura 23 "Correu externalitzat a Gmail" es pot veure com l'Escola del Treball de Barcelona fa anys que disposa del seu correu externalitzat i gestionat per *gmail*. Els seus usuaris utilitzen adreces de correu del tipus "*user@escoladeltreball.org*" però en realitat qui s'encarrega de la gestió és l'empresa *Google*.

Figura 23. Correu externalitzat a Gmail.



1.7.Utilitza la signatura digital i el correu xifrat.

Les comunicacions per email s'han tornat cada vegada més importants en la vida diària, no només pels emails amb acudits, presentacions gràcies o crítiques al govern. També són imprescindibles pel funcionament i comunicació de moltíssimes empreses i organitzacions. De fet, molta documentació que abans es feia per escrit ara es fa telemàticament per email. Aixó implica la necessitat de poder estar segur de l'identitat de l'emissor i del receptor dels emails. En entorns de seguretat més exigents fins i tot es pot requerir el xifrat del contingut per evitar que sigui accessible per tercers.

El concepte clau per a la confiança en les comunicacions per email és la firma digital que permet assegurar que un emissor és qui diu ser i garanteix també que el contingut del missatge no s'ha alterat per tercers. Per implementar la signatura digital i el xifrat cal utilitzar certificats digitals.

Primerament es farà un repàs als conceptes generals de seguretat i es comentaran superficialment tots aquells termes relacionats amb la seguretat que hem sentit a vegades i que no acabem d'ubicar ben bé quina funció fan. També es descriuran clàrament els puntals de la seguretat en el correu: autenticació, integritat, no repudi i xifrat. S'aprendrà a crear parelles de claus públiques i privades (els famosos certificats digitals) i es tractaran els diversos formats existents.

!!

Als annexos trobareu una completa explicació de conceptes globals de seguretat i també annexos específics per a PGP, S/MIME i certificats digitals

El MTA i els servidors de correus transporten missatges independentment del fet de que estiguin xifrats i/o signats. Són els clients de correu els que han de proporcionar aquesta capacitat a l'usuari.

Per poder gestionar correu signat i xifrat cal utilitzar clients de correu que permetin fer aquestes funcions i cada usuari ha de disposar dels certificats digitals apropiats. Per al protocol de transport de correu SMTP i el servidor de correu (per exemple *sendmail*) que el missatge que processen sigui xifrat i signat li és transparent, indiferent, igual que per al carter no li implica cap funcionament diferent el fet que el contingut d'una carta estigui xifrat amb una clau secreta o que la carta porti imprès el segell d'una entitat. Són els clients de correu com per exemple el *thunderbird* els que han d'incorporar la capacitat de gestionar aquest

tipus de correu. En el cas del *thunderbird* caldrà afegir-li software adicional per poder signar i xifrar missatges usant Open PGP o S/MIME.

Les dues tecnologies de correu signat i xifrat que es treballaran són:

- **Open PGP:** Aquesta tecnologia permet la generació de missatges de correu xifrats i/o signats i evidentment desxifrar el contingut i verificar les signatures. Es basa en el programa original PGP 'Pretty Good Privacy' desenvolupat per Phil Zimmermann. Utilitza el model de seguretat anomenat "*web of trust*" on cada usuari és el responsable de la gestió de la confiança amb els certificats dels altres usuaris.
- **S/MIME:** Aquest estàndard proporciona les mateixes característiques de signatura digital i xifrat (i a la inversa) que proporciona Open PGP però requereix un altre format per als certificats digitals. El model de seguretat que utilitza és el model PKI o *Public Key Infrastructure* on existeix una estructura piramidal d'entitats de certificació o CA (Certification Authority) que determinen la confiança amb els certificats digitals. Aquest model PKI és el model en el que es basen la majoria de protocols de seguretat d'internet com HTTPS, SMTPS, etc que de fet utilitzen SSL o TLS. Els navegadors web venen usualment pre-carregats amb els certificats de les CA més importants a nivell global d'internet.

1.7.1. Mecanismes de seguretat en el correu.

L'intercanvi de missatges de correu es produeix utilitzant protocols SMTP, POP i IMAP que són protocols insegurs, és a dir, el seu contingut viatja en forma de text pla. Per tant, qualsevol intermediari en la xarxa pot monitoritzar (*sniffer*) el contingut dels missatges. Qualsevol eina tipus *wireshark* permet fer un seguiment dels continguts de qualsevol protocol TCP en text pla.

Monitoritzar contingut

Un exemple típic d'aquest fet és quan amb el professor es monitoritzen els diàlegs entre dos hosts de classe (o de casa) usant *wireshark*. A part de poder fer un seguiment de les trames utilitzant l'opció de seguiment del fluxe TCP anomenada *Follow tcp stream* es pot observar clarament el 'text' de tot el diàleg efectuat.

Això significa que quan dos interlocutors intercanvien missatges per qualsevol xarxa el contingut del seu diàleg pot ser monitoritzat per tercers? Si, evidentment!. No només això sinó que la conversa pot ser falsejada per aquests tercers (el famós *man-in-the-middle*). És a dir, un usuari creu estar dialogant amb la seva entitat bancària i en realitat ho està fent amb uns 'altres caradures'.

Propietats de seguretat

Tot seguit s'analitzen els aspectes de seguretat que són desitjables en la comunicació entre dos interlocutors:

- Confidencialitat (Xifrat).
- Autenticitat.
- Integritat.
- No-repudi.

Un error típic es barrejar aquests conceptes i pensar que van tots tres a la una, conjuntament, i no és així. Cal aplicar per a cada cas aquell tipus de seguretat que faci falta.

Confidencialitat (Xifrar)

Si un emissor vol enviar un missatge 'secret' a un receptor de manera que únicament el receptor el pugui entendre cal un mecanisme de xifrat. Xifrar és codificar el missatge utilitzant algun tipus de clau o mecanisme de xifrat per fer inaccessible el missatge a qualsevol altre usuari excepte al destinatari. El destinatari ha de conèixer la clau o ha de disposar d'un mecanisme per desxifrar el missatge i obtenir-ne el contingut original.

Xifrar és codificar el missatge utilitzant algun tipus de clau o mecanisme de xifrat per fer inaccessible el missatge a qualsevol

altre usuari excepte al destinatari.

El fet de que el missatge estigui xifrat garanteix que només l'emissor és capaç d'entendre'n el contingut (si suposem que han acordat el mecanisme de desxifrat). Ara bé, pot estar segur el destinatari de que el missatge prové realment de qui creu que prové? Pot estar segur de que el missatge no ha estat alterat?.

Veurem que xifrar un missatge proporciona **confidencialitat** però no és un mecanisme que proporcioni seguretat de que l'emissor és qui diu ser ni de que el missatge és tal i com era en l'origen (sense modificacions de tercers).

Autenticitat

Un receptor rep un missatge del banc informant-lo que ha de fer un pagament, o que ha d'enviar el número de la seva targeta de crèdit per tal o qual propòsit. Pot estar segur el receptor que el missatge que rep procedeix realment del banc?

L'**autenticació** és la característica de seguretat que permet a un receptor estar segur que el missatge prové de qui diu provenir. No només al receptor sinó que també proporciona a l'emissor la garantia de que el receptor 'sap del cert' que el missatge li ha enviat ell.

No-repudi

Si un emissor envia un missatge autenticat a un emissor no hi ha manera legal de desdir-se de que l'ha enviat. Imagineu que un directiu d'una empresa envia un missatge inapropiat a un empleat, si el missatge és autenticat el directiu no pot negar legalment que l'ha enviat. El mateix pot passar si l'empresa A envia un email a un client oferint-li els productes a meitat de preu i després intenta retractar-se'n dient que l'email no era seu. Si el missatge és autenticat queda legalment demostrat que l'empresa A ha sigut l'emissor.

Aquesta característica que proporciona l'autenticació que impedeix que l'emissor pugui negar que ha enviat el missatge s'anomena **no repudi**.

Caldrà doncs un mecanisme de seguretat que permeti a un emissor enviar missatges de manera que els receptors tinguin la certesa absoluta de que l'emissor és qui diu ser.

Una confusió habitual és creure que per establir seguretat també cal xifrar. Això no és cert. Una administració pública, per exemple, vol poder enviar missatges als ciutadans garantint l'autenticitat del missatge, però no li cal xifrar el missatge, no li cal que siguin secrets.

Integritat

N'hi ha prou amb el xifrat i l'autenticació per establir una comunicació cent per cent segura?. No. Fixeu-vos que si una entitat bancària envia un email ordenant un pagament de 1000 euros i un tercer té la capacitat de poder modificar el missatge i posar-hi un parell de zeros més (com qui no vol la cosa...) llavors s'estaria ordenant un pagament de deu mil euros!. Què és el que ha fallat aquí? Que el missatge ha estat modificat per tercers. El missatge podria procedir de qui diu que procedeix i fins i tot podria estar xifrat, però algú molt espavilat ha aconseguit colar-hi un parell de zeros més.

La **integritat** és la propietat de seguretat que garanteix que el missatge no ha estat alterat, que arriba al destinatari tal i com s'ha generat en l'origen.

Per implementar integritat no cal xifrar els missatges però veurem que la integritat i l'autenticitat es proporcionen conjuntament.

Implementar seguretat

Per implementar seguretat en la comunicació de missatges avui en dia s'utilitza abastament els mecanismes de certificats digitals basants en

criptografia de clau asimètrica. És a dir, basants en que cada interlocutor disposa d'una clau privada (coneguda i accessible només per ell) i d'una clau pública o *certificat* coneguda per tots els interlocutors. Els certificats són les claus públiques *signades*, avalades, per una entitat de certificació o CA (Certification Authority).

Per implementar seguretat els interlocutors disposen de:

- una **clau privada**
- un **certificat** o clau pública signada per una entitat CA.

En realitat tot el 'muntatge' de la segurtat és més complexe si s'hi afegueixen les CA, els anells de clau, el sistema PKI, etc. Però en aquest dossier intentarem mantenir el més simplificat possible tot aquest model i usar el que sigui estrictament necessari per a la comunicació segura entre dos interlocutors.

Tot següent s'analitzarà el funcionament global que permet implementar la seguretat de clau pública/privada:

- Xifrat.
- Signatura o Certificat Digital.

Per entendre els apartats següents cal tenir molt present que la clau privada és privada, només la coneix i només hi pot accedir el seu propietari. Mai es comunica a un altre i mai un altre la pot usar. En canvi la clau pública és coneguda per tothom, de fet si els altres no la coneixen cal enviar-la per tal de que els altres la tinguin disponible. Si no fos així el mecanisme de clau pública/privada no funcionaria.

La parella clau pública/privada permeten xifrar i signar però en el sentit de que quan se n'utilitza una per fer una acció, cal l'altre per desfer la acció. Functionen per parelles, si una fa l'altre desfa, però també a l'inrevés.

Mala interpretació del funcionament de les claus.

Un error típic és creure que cada clau només pot fer una cosa i dir 'la clau privada sempre xifra i signa i la pública desxifra i verifica'. Doncs NO. No funcionen així. Segons l'acció a fer se n'utilitza una o l'altre (i per desfer l'acció sempre cal aplicar l'altre).

Xifrat

Un *emissor-A* vol enviar un missatge *msg* xifrat a un *destinatari-B*, de manera que únicament el destinatari tingui la capacitat de saber el contingut real del missatge. Què cal fer? Pensem.

L'*emissor-A* vol xifrar el missatge de manera que únicament el *destinatari-B* el pugui entendre. Quina és la 'cosa' que únicament el *destinatari-B* té i que la resta del món no té? Allò que ningú més que *destinatari-B* pot tenir? La clau privada del *destinatari-B*.

Així doncs l'emissor envia el missatge xifrat amb la clau privada del destinatari? NO, IMPOSSIBLE. L'emissor no pot fer això perquè la clau privada del destinatari només la coneix el destinatari.

Què coneix l'emissor del destinatari? La clau pública del destinatari. De manera que l'emissor xifra el missatge amb la clau pública del destinatari. De fet qualsevol emissor del món ho pot fer perquè la clau pública del *destinatari-B* és precisament pública.

Un cop xifrat el missatge pot la resta del mon (inclòs l'emissor) desxifrar-lo? No. Únicament el pot desxifrar qui tingui la clau privada associada a la clau pública usada per xifrar. És a dir, només podrà desxifrar el missatge qui disposi de la clau privada *destinatari-B*, o sigui només el destinatari podrà desxifrar el missatge.

Qualsevol pot **xifrar** un missatge usant la **clau pública del destinatari**, que precisament és pública. Únicament el destinatari pot **desxifrar** el missatge usant **la seva pròpia clau privada**, que només ell té.

Signar

Signar un missatge proporciona **integritat**, **autenticació** i **no repudi** simultàniament. Quan un missatge va signat per l'emissor el missatge és irrefutable que procedeix d'ell i a més a més es garanteix que no s'ha modificat per tercers. El receptor ha de poder **verificar** que el missatge és autèntic.

El procés físic de signar el missatge consisteix en afegir al missatge el certificat digital de l'emissor i és per això que de vegades el fet se signar

Publicar la clau pública

La clau pública ha de ser coneguda per els altres interlocutors. Els mecanismes per donar-la aconèixer són:

- publicar-la en un servidor públic de claus.
- enviar la clau als 'coneguts'.
- adjuntar la clau pública amb els propis missatges.

un missatge s'anomena certificar-lo o es parla de missatge amb certificat digital.

Anem a pams. Com es signa un missatge? Un *emissor-A* vol garantir a un *destinatari-B* que el missatge l'ha enviat ell i únicament ell, i que no s'ha modificat per tercers. Com ho pot fer?. Què té l'emissor que només tingui ell i ningú més del món? i que demostra que és ell. La seva clau privada. L'*emissor-A* signa el missatge utilitzant la seva clau privada. Això només ho pot fer ell ja que ningú més té aquesta clau privada.

El receptor del missatge quan el rep ha de poder verificar que és realment de l'*emissor-A*. Quina cosa té el *destinatari-B* de l'emissor? El destinatari disposa de la clau pública de l'emissor. De fet qualsevol destinatari del missatge pot verificar-ne la signatura perquè la clau pública de l'emissor és precisament pública.

Així, doncs, el receptor verifica la signatura utilitzant la clau pública de l'emissor. Com que les parelles de claus pública/privada funcionen només conjuntament, la verificació només serà correcta si el missatge s'ha xifrat amb la clau privada corresponent. És a dir, la verificació amb la clau pública de l'*emissor-A* només funcionarà si el missatge s'ha signat amb la clau privada de l'*emissor-A*.

Per **signar** un missatge l'emissor utilitza la **seva pròpia clau privada**, que només té ell.

Per **verificar** un missatge el receptor utilitza la **clau pública de l'emissor**, disponible per a tothom.

La verificació només serà correcta si el missatge s'ha signat amb la clau privada que és la parella de la clau pública.

Altres cops observar que el fet de que el missatge estigui signat no significa que el missatge sigui secret. Per ser secret ha d'anar també xifrat.

El procés mecànic de signar un missatge consisteix en aplicar la clau privada al missatge. Es fan els següents processos:

- **Integritat:** De fet no es codifica tot el missatge amb la clau privada, ja que implica un sobrecost de temps i d'esforç de càlcul. El procés tècnic que es fa és generar un **hash** o resum del missatge i signar aquest *hash*. Si el missatge es modifica per tercers el hash no coincidiria amb el missatge.
- **Autenticitat:** per autenticar el missatge s'adjunta el certificat o clau pública de l'emissor avalat per una CA o autoritat de certificació. Aquest certificat i el *hash* van codificats amb la clau privada. No tot el missatge, només aquesta part. Així el receptor verifica amb la clau pública de l'emissor que el certificat de l'emissor és vàlid i que el hash també.

Tècniques de hash o resum

Les tècniques de *hash* o 'resum' són molt habituals en el món de la informàtica. Per exemple segur que heu observat que molts passwords es codifiquen amb MD5 corresponents al format *Message Digest 5*. La generació de 'resums' o *Message Digest* és un mecanisme que a partir d'un contingut com per exemple un text o un missatge genera un seqüència de n-bytes (per exemple 128) que és única.

Se suposa que donat un 'resum' d'un contingut no hi pot haver-hi cap altre contingut en l'univers que generi el mateix 'resum'. Si el contingut es modifica és impossible que generi el mateix 'resum'.

També es garanteix que partint del 'resum' és impossible obtenir el contingut original. De fet això es ben lògic quan 1000 pàgines d'un text són resumides en un simple *hash* de 128 bytes.

Es resum, podem dir que en el procés de signar un missatge o incorporar al missatge una 'firma digital' consta dels passos següents:

- genera un *hash* o *message digest* o *fingerprnt* del missatge.
- es xifra el *hash* amb la clau privada de l'emissor. Això és una signatura digital.
- el destinatari rep el missatge i calcula un nou hash en el receptor, és a dir, calcula de nou el hash basant-se amb el contingut del missatge rebut.
- la signatura digital rebuda (hash xifrat) es desxifra amb la clau pública de l'emissor.
- els dos hash han de ser iguals, això garanteix l'autenticació i la integritat.
- garanteix l'autenticació: només l'emissor pot haver generat el missatge si és desxifrabla per la clau pública de l'emissor.
- garanteix integritat: ningú més pot haver modificat el missatge perquè això hagués modificat el hash i ningú més té la clau privada de

l'emissor per tornar-lo a signar.

- avantatges: no cal xifrar tot el missatge, només el hash, que és la part que garanteix que no s'ha modificat.

Publicar les claus públiques

Un dels fets cabdals en l'utilització de la criptografia asimètrica de clau pública / clau privada és que les claus públiques dels usuaris han d'estar disponibles per als altres usuaris, les han de tenir 'abans' de poder realitzar les comunicacions xifrades. És a dir, un usuari s'ha de preocupar de 'propagar' la seva clau pública a els altres usuaris amb els que vol poder mantenir comunicacions segures.

Els principals mecanismes per fer conèixer la pròpia clau pública a altres destinataris són:

- **Enviar** la clau pública a aquells destinataris amb els que es vol poder mantenir comunicacions segures. Entre dos usuaris cal que s'intercanviïn les seves claus públiques. Entre tres cal que cada un disposi de la pública dels altres dos, etc. És a dir, cal preocupar-se de tenir o fer arribar (depèn del punt de vista) la clau pública dels altres.
- **Adjuntar** la clau pública en els missatges de correu 'no segurs' que un usuari envia als altres destinataris. És a dir, adjuntar la clau en tots els missatges. D'aquesta manera els destinataris habituals poden anar incorporant la clau pública al seu anell de claus i poder passar a mantenir comunicacions 'segures'. Un dels principals problemes és que es usuaris inexperiments no saben què fer amb les claus que s'adjunten en els missatges normals.
- **Publicar en servidors de claus** les claus públiques d'un usuari. Existeixen a internet servidors de claus públiques coneguts on qualsevol usuari pot publicar les seves claus. Així els altres usuaris en lloc d'aconseguir-les a través d'un intercanvi directe les aconsegueixen a través d'un tercer o intermediari, el servidor públic de claus.

La manera com s'implementa la confiança entre els certificats s'anomena "**model de seguretat de confiança**". Segons sigui l'eina utilitzada per implementar correu segur *OpenPGP* o *S/MIME* s'utilitza un model de seguretat de confiança o un altre. Els dos models actuals són:

- **PKI** *Public Key Infrastructure* és el model on la confiança amb les claus públiques o certificats es basa en una estructura piramidal d'entitats de certificació o CA. Les entitats de certificació CA 'avalen' els certificats amb la seva garantia o confiança. Els altres usuaris confien en els certificats si han sigut emesos per entitats en les que confien. Existeixen entitats de nivell global a internet en les que 'confia tothom' o si més no els navegadors web venen pre-carregats amb aquestes entitats com a entitats de certificació vàlides. D'altres entitats poden pretendre avalar-se per si soles o poden ser sub entitats avalades per una CA de rang superior. En tot cas en un certificat ha de poder-se avaluar examinant la cadena de confiança de l'emissor del certificat.
- **Web of trust** En aquest model de confiança, anomenat també 'confiança entre parells o entre iguals' no existeixen autoritats de certificació ni una estructura piramidal per 'avaluar' certificats. Cada usuari és el responsable de decidir si confia o no en certificats d'altri. En aquest model els usuaris s'intercanvien els certificats els uns i els altres i hi confien segons el seu pròpi criteri. Aquest model permet estendre la confiança a tercers, és a dir, si "pere" confia en "anna" l'usuari "pere" pot decidir (o no) confiar també amb tothom amb qui confii "anna".

1.7.2.GPG: Seguretat en consola.

Com implementar seguretat en els correus? Com enviar missatges de correu xifrats i signats? Com verificar que l'emissor d'un correu és qui diu

ser i que el missatge no ha estat modificat per tercers?

Un software ampliament usat per implementar seguretat és **PGP Pretty Good Privacy**. Existeix una versió feta per GNU de prestacions similars anomenada **GNU PG** o més popularment **GPG**. Aquest és el software que s'utilitzarà en aquests exemples.

El software de PGP va ser desenvolupat el 1991 per Peter Zimmermann que va acabar fugint dels Estats Units perquè al seu govern no li va fer gens de gràcia que 'algú' implementés un mecanisme públic per poder xifrar continguts. De fet sembla que aquest senyor es va passar 15 anys fugint del FBI. Si voleu podeu consultar a la *wikipèdia* més abastament les seves peripècies i la història del software PGP.

De fet al govern dels Estats Units sembla que no sempre li agraden els avenços en matèria de software de seguretat. 'Es diu' que quan *Netscape* va implementar comunicacions HTTP segures usant SSL el xifrat original era de 64 Bytes, però el govern va obligar a reduir-ho a 56 Bytes. 'Es diu' que el motiu d'aquesta imposició és que el govern no podia desxifrar els continguts xifrats a 64 Bytes però si els xifrats a 56 Bytes. O sigui que això de comunicacions segures...

Els següents exemples mostraran com enviar emails xifrats i signats en entorn de consola utilitzant l'ordre **email**. De fet més que de la utilització de l'ordre *email* es tracta de saber transformar contingut text o de qualsevol altre mena, en forma de fitxers, i generar un nou contingut xifrat, signat o totes dues coses.

Preparació dels usuaris i continguts

Per poder exposar el funcionament del correu segur es crearan dos usuaris "pere" i "anna" que s'intercanviaran missatges. Per cada usuari es generarà la parella de claus pública/privada, la privada serà inaccessible per els altres i la clau pública es farà arribar a l'altre interlocutor.

```
# Generar els usuaris
[root@host ~]# useradd pere
[root@host ~]# passwd pere
[root@host ~]# useradd anna
[root@host ~]# passwd anna

# Verificar que es disposa del software GNU PG
# rpm -q gnupg
gnupg-1.4.10-1.fc12.i686
```

Per a cada usuari cal generar la parella de claus pública/privada. Aquest procediment interactiu consta de vàries passos:

- Es genera un directori de nom *.gnupg* en el home de l'usuari on es desarà tot allò regerent al programa *gpg*.
- Es genera un fitxer *gpg.conf* que contindrà la configuració del servei de seguretat.
- Es crea un fitxer per a l'anell de claus secretes anomenat *secring.gpg* i un fitxer amb l'anell de claus públic *pubring.gpg*.
- Escollir el tipus de clau a usar. Es pot escollir el tipus per defecte, que ha de permetre xifrar i signar.
- Indicar la mida en bits de la clau. El rang de bits a escollir varia segons la clau a usar. Es pot escollir el valor per defecte.
- Establir la data de caducitat de la clau. El valor per defecte '0' indica que la clau no caduca mai.
- Introduir la informació descriptiva del compte de l'usuari a qui pertany la clau. Cal indicar el nom, email i un comentari descriptiu.
- Assignar el password a la clau privada.
- Generar entropia en el sistema per tal de que el software de GPG pugui generar una clau suficientment aleatòria.

Per generar la parella de claus cal indicar l'ordre *gpg -gen-key* i escollir el tipus de clau a utilitzar. En l'exemple s'ha escollit el tipus per defecte. Els dos primers permeten tant xifrar com signar. Els dos últims només signar.

```
# Generar la parella de claus
[pere@host ~]$ gpg --gen-key
gpg (GnuPG) 1.4.10; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory `/home/pere/.gnupg' created
gpg: s'ha creat el nou fitxer d'opcions «/home/pere/.gnupg/gpg.conf»
gpg: AVÍS: les opcions en «/home/pere/.gnupg/gpg.conf» encara no estan actives durant aquesta execució
gpg: s'ha creat l'anell «/home/pere/.gnupg/secring.gpg»
gpg: s'ha creat l'anell «/home/pere/.gnupg/pubring.gpg»
Seleccioneu quin tipus de clau voleu:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (només signar)
(4) RSA (només signar)
La vostra selecció? 1
```

Cal escollir la mida en bites de la clau. El valor per defecte escollit és 2048. Com més bits s'utilitzin més bona és la clau, més difícil de ser

desxifrada per un atacant, però també més costosa de manipular. És a dir, més bits vol dir més temps de càlcul en processar-la.

```
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
La grandària sol·licitada és 2048 bits
Especifiqueu el temps de validesa de la clau.
  0 = la clau no caduca
  <n> = la clau caduca als n dies
  <n>w = la clau caduca a les n setmanes
  <n>m = la clau caduca als n mesos
  <n>y = la clau caduca als n anys
Indiqueu la validesa de la clau (0)
Key does not expire at all
Is this correct? (y/N) y
```

Introduïr la informació descriptiva del compte de l'usuari a qui pertany la clau. Cal indicar el nom, email i un comentari descriptiu. A continuació cal introduir el password que permet bloquejar (xifrar) el fitxer de la clau privada. El fitxer de la clau privada es pot guardar tal qual amb text pla (llavors cal assegurar-se de posar-li els permisos apropiats per impedir-ne l'accés d'altres) o xifrat de manera que encara que d'altres puguin llistar-ne el contingut no serveix de res perquè està xifrat. És molt bona idea protegir el fitxer de clau privada amb aquesta encriptació simètrica basada en una paraula clau o password.

L'avantatge de xifrar la clau privada és que la protegeix contra accessos indeguts. El desavantatge és que cada vegada que cal accedir-hi cal introduir el password de desxifrat del fitxer.

```
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
  "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Nom i cognoms: pere pou prat
Adreça electrònica: pere@localhost.localdomain
Comentari: compte d'en pere local a la màquina
Esteu usant el joc de caràcters 'utf-8'.
Heu triat l'identificador d'usuari:
  "pere pou prat (compte d'en pere local a la màquina) <pere@localhost.localdomain>"

Canvia (N)om, (C)omentari, (E)mail o (O) d'acord / (X) ix 0
Cal una contrasenya per a protegir la clau secreta.

Introduïu la contrasenya: privadapere
Repetiu la contrasenya: privadapere
```

Generar entropia en el sistema per tal de que el software de GPG pugui generar una clau suficientment aleatòria.

```
Cal generar molts bits aleatòriament. És bona idea fer alguna altra cosa
(teclejar, moure el ratolí, usar els discos) durant la generació de
nombres primers; açò dóna oportunitat al generador de nombres aleatoris
d'aconseguir prou entropia.
...+++++
...+++++
gpg: /home/pere/.gnupg/trustdb.gpg: s'ha creat la base de dades de confiança
gpg: key 3CBDFE49 marked as ultimately trusted
s'han creat i signat les claus pública i secreta.

gpg: s'està comprovant la base de dades de confiança
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/3CBDFE49 2012-01-28
    Key fingerprint = F881 A4A8 F131 86DB F696 A02B 8D27 B681 3CBD FE49
uid pere pou prat (compte d'en pere local a la màquina)
    <pere@localhost.localdomain>
sub 2048R/CE033F03 2012-01-28
```

Cal repetir el mateix procés per als altres usuaris, en l'exemple per a la usuària "anna". Un cop creada la clau privada observar l'estructura de directòris i fitxers implicats:

```
[pere@host ~]$ tree .gnupg/
.gnupg/
├── gpg.conf
├── pubring.gpg
├── pubring.gpg~
├── random_seed
├── secring.gpg
└── trustdb.gpg
```

Verificació de les claus

Un cop creades les claus es poden consultar amb les ordres que permeten llistar les claus públiques i les claus privades. El següent codi llista les claus públiques de l'usuari "pere" i a continuació les claus privades.

```
# Llistat de les claus públiques
[pere@host ~]$ gpg --list-keys
/home/pere/.gnupg/pubring.gpg
-----
pub 2048R/3CBDFE49 2012-01-28
uid pere pou prat (compte d'en pere local a la màquina)
    <pere@localhost.localdomain>
sub 2048R/CE033F03 2012-01-28
```



```
# Llistat de les claus privades
[pere@host ~]$ gpg --list-secret-keys
/home/pere/.gnupg/secring.gpg
-----
sec   2048R/3CBDFE49 2012-01-28
uid           pere pou prat (compte d'en pere local a la màquina) <pere@localhost.localdomain>
ssb   2048R/CE033F03 2012-01-28
```

Els pot observar el *fingerprint* o **empremta** d'una clau amb l'opció *-fingerprint*.

```
[pere@host ~]$ gpg --fingerprint 3CBDFE49
pub   2048R/3CBDFE49 2012-01-28
      Key fingerprint = F881 A4A8 F131 86DB F696 A02B 8D27 B681 3CBD FE49
uid   pere pou prat (compte d'en pere local a la màquina)
      <pere@localhost.localdomain>
sub   2048R/CE033F03 2012-01-28
```

Utilització de GPG

Els exemples següents mostraran com xifrar, signar i xifrar+signar un contingut abans de ser enviat amb l'ordre email. El contingut pot ser qualsevol fitxer però per simplicitat s'ha optat per usar un fitxer de text pla fàcil d'identificar.

El següent codi es presenta a tall de 'xuleta' de les instruccions que s'utilitzaran al llarg dels exemples:

```
Generar claus:
$ gpg --genkey
$ gpg --list-keys
$ gpg --list-secret-keys

Esborrar claus:
$ gpg --delete-key clauID
$ gpg --delete-secret-key clau ID
$ gpg --gen-revoke ID

Importar/Exportar:
$ gpg --armor --output fileout --export ClauID
$ gpg --armor --output fileout --export-secret-key
$ gpg --import public.key

Xifrar:
$ gpg --armor --output fileout --encrypt file
$ gpg --armor --output fileout --recipient user@domain --encrypt file
$ gpg --output fileout --decrypt file-encrypted

Signar:
$ gpg --clearsign file
$ gpg --sign file
$ gpg --decrypt file.gpg
$ gpg --verify file-signed
$ gpg --detach-sign

Xifrar + Signar
$ gpg --armor --recipient user@host --sign --encrypt file
$ gpg --armor --recipient user@host --encrypt --sign file

Enviar:
$ mail -s "msg" destinatari < file
$ gpg ...opcions... | mail -s "msg" destinatari

Signatures de confiança:
$ gpg --list-sigs
$ gpg --sign-key
```

Exemple 01: xifrar/dessifrar un contingut

En aquest exemple es xifra un fitxer de nom "carta.txt" generant-ne un de nou xifrat. El procés es realitza tal i com es mostra en el següent llistat de codi.

```
# contingut del fitxer a xifrar
[pere@host ~]$ cat carta.txt
aquest és un exemple de text pla que podria
ser un missatge.
L'ha escrit l'usuari pere i serà usat per
xifrar i per signar en altres varis exemples.
Bye

# Xifrar el fitxer
[pere@host ~]$ gpg --armor --output carta.xifrat.txt --encrypt carta.txt
No heu especificat un ID d'usuari. (podeu usar «-r»)
Current recipients:
Introduïu l'ID d'usuari. Finalitzeu amb una línia en blanc: 3CBDFE49
Current recipients:
 2048R/CE033F03 2012-01-28 "pere pou prat (compte d'en pere local a la màquina)
 <pere@localhost.localdomain>"
Introduïu l'ID d'usuari. Finalitzeu amb una línia en blanc:

# Observar els fitxers
```

```
[pere@host ~]$ ll
-rw-rw-r-- 1 pere pere 156 28 gen 16:41 carta.txt
-rw-rw-r-- 1 pere pere 722 28 gen 16:47 carta.xifrat.txt

# Observar el fitxer xifrat
[pere@host ~]$ cat carta.xifrat.txt
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.10 (GNU/Linux)

hQEMAx1zqKP0Az8DA0gAp3Vf9b2tbXXrSh/2/H00G/3+WZLxQwJLdjIDHXeDQbY3
oSGVvN3PQ7WMyxmojvvgmN54b7Gu4xeTgxZASjRfKU0hZTzsA9VYd1rfTsU+ViDk
nQTVRCcCch6vV4NuzpsJceeZ6xWaK75IxgpM3IL4hR7B414Vwxe2QK9VmshA6q4J
xSB2iNYU+15X505G0fhs+GxrmzZrZtNIxi8wXk0VtDDA73f86b422hP3lFFe5DYx
TQwJLRik/OgWG33qRrj140kuXma/HRUCGLPaveSVIwF+Qs9ZhnJNu+VxiNtwXWYf
g+Ebs/RnrA5Pz/j9cv3o3849GUomMFrFRI20Z0qtRtK8AYvLDHaHjIQJ44n0duHh
PwUev+6l1ovxLCh01yP11N+LquDmmHiQGPdHYh8ReB1dtZtysamqGzkSfcZc+m9Q
xuxhEjPM2NhtsrVxHeIpOJJAs8TZRN/gUi/hLY2eX5MPLrTzwUPjtx6Ag8KXnjH0
y7y+ximCaZPyY+xam3qZ8680+XsX0vHJsdocSY2bHInkIbrobTX+huJ3Bf09+wf
9cZWaXo7D1hBD06huVXIF2Sw0wzBpgA0Cg1TIfc=
=Qjrw
-----END PGP MESSAGE-----
```

L'ordre efectuada per ecriptar utilitza les opcions:

- *-armor* indica que el resultat generat ha de ser un fitxer de text i no un fitxer binari.
- *-encrypt* indica el nom del fitxer a encriptar.
- *-output* permet indicar el nom que ha de tenir el fitxer encriptat. Si no s'indica s'utilitza stdout.

Un seguimet de les accions realitzades mostra que:

- Primerament es genera el contingut del fitxer.
- Tot seguit s'invoca l'ordre d'encriptació.
- Cal indicar l'UID de la clau pública a usar per a la encriptació, la del destinatari. Es pot indicar interactivament o com argument en la línia d'ordres. En aquesta ocasió s'ha utilitzat l'UID de la clau pública del pròpi usuari "pere".
- Interactivament es tornen a demanar més UIDs de clau pública corresponents a altres destinataris. Si es prem enter finalitza.
- Observar el contingut xifrat. És un fitxer de text ascii pla amb el contingut xifrat. Es pot observar la capçalera i peu PGP.

Un cop xifrat el missatge es fa arribar al destinatari i aquest el pot desxifrar amb la seva pròpia clau privada. En aquest exemple 'simple' el missatge l'ha codificat l'usuari "pere" destinat a ell mateix (una mica absurd però suficient per començar a practicar). Així doncs, ara cal que "pere" desxifri el missatge amb la seva clau privada:

```
# Desencriptar el fitxer encriptat
[pere@host ~]$ gpg --output carta.desxifrada.txt --decrypt carta.xifrat.txt

You need a passphrase to unlock the secret key for
user: "pere pou prat (compte d'en pere local a la màquina) <pere@localhost.localdomain>"
2048-bit RSA key, ID CE033F03, created 2012-01-28 (main key ID 3CBDFE49)

gpg: encrypted with 2048-bit RSA key, ID CE033F03, created 2012-01-28
      "pere pou prat (compte d'en pere local a la màquina) <pere@localhost.localdomain>"

# Llistar els fitxers generats (original, xifrat i desxifrat)
[pere@host ~]$ ll
-rw-rw-r-- 1 pere pere 156 28 gen 17:26 carta.desxifrada.txt
-rw-rw-r-- 1 pere pere 156 28 gen 16:41 carta.txt
-rw-rw-r-- 1 pere pere 722 28 gen 16:47 carta.xifrat.txt

# Comprovar que carta.txt i carta.desxifrada.txt són iguals
[pere@host ~]$ diff carta.txt carta.desxifrada.txt
```

Un seguiment de les accions realitzades permet observar que per desxifrar s'han usat les opcions:

- *-decrypt* que permet indicar el nom del fitxer a usar per desencriptar.
- *-output* indica el nom del fitxer a generar amb la sortida desencriptada. Si no s'indica es realitza per stdout.
- per poder usar la clau privada del receptor ha calgut introduir la contrassenya del fitxer de la clau privada (establerta a "privadapere").
- *recipient* (aquesta opció no s'ha usat) permet escollir quina de les claus es vol usar per el email en lloc de per el ID.

Exemple 02: Intercanvi de claus

En l'exemple anterior s'ha vist com xifrar i desxifrar un fitxer, però s'ha fet tot des del mateix usuari. Evidenment això no té cap mena de substància. El propòsit del PGP és permetre que usuaris diferents s'intercanviïn missatges xifrats i/o signats.

Per poder fer que l'usuari "pere" envii un missatge a l'usuària "anna" cal que l'emisor disposi de la clau pública del destinatari. Caldrà aprendre a exportar, importar i esborrar claus de l'anell de claus.

Existeix un anell de claus per a les claus públiques anomenat *pubring.gpg*. En aquest fitxer s'acumulen les claus públiques conegudes pel GPG. Un anell de claus és un clauer. A mida que acumulem claus de casa, de la feina, del garatge, etc, les anem afegint en un clauer. El GPG fa el mateix i a mida que va coneixient claus públiques les va incorporant al clauer.

L'anell de claus privades es troba en un fitxer anomenat *secring.gpg*. Si les claus privades només les pot tenir el propi usuari fa falta un anell de claus? Pot tenir més d'una clau privada?. Si, igual que passa amb la clau de la porta de l'edifici, la de la porta de casa, la de etc. Un usuari pot generar múltiples claus privades que utilitza per a propòsits diferents. Així per exemple l'usuari "pere" pot tenir una clau privada per les seves transaccions econòmiques privades, una altra utilitzada per a les coses de feina, una altra per a la seva 'vidilla' privada, etc.

En el següent llistat de codi es pot veure com l'usuari "pere" genera un parell de claus més per a propòsits diferents, una per a "operacions bancàries" i una altra per a "compres per internet". En el codi es poden observar els ID de cada clau pública: 3CBDFE49, 42B08992 i 6D17F7B.

```
# Generar la clau per a operacions bancàries
[pere@host ~]$ gpg --genkey
...
Nom i cognoms: pere pou prat
Adreça electrònica: pere@localhost.localdomain
Comentari: clau per a operacions bancàries
...

# Generar la clau per a compres d'internet
[pere@host ~]$ gpg --genkey
...
Nom i cognoms: pere pou prat
Adreça electrònica: pere24@yahoo.com
Comentari: clau per a compres d'internet
Heu triat l'identificador d'usuari:
...

# Llistar les claus de l'usuari "pere"
[pere@portatil ~]$ gpg --list-keys
/home/pere/.gnupg/pubring.gpg
-----
pub  2048R/3CBDFE49 2012-01-28
uid  pere pou prat (compte d'en pere local a la màquina)
     <pere@localhost.localdomain>
sub  2048R/CE033F03 2012-01-28

pub  2048R/42B08992 2012-01-28
uid  pere pou prat (clau per a operacions bancàries)
     <pere@localhost.localdomain>
sub  2048R/89B0F5C4 2012-01-28

pub  2048R/76D17F7B 2012-01-28
uid  pere pou prat (clau per a compres d'internet)
     <pere24@yahoo.com>
sub  2048R/D5B5D027 2012-01-28
```

Per exportar i importar claus públiques i privades la sintàxis a utilitzar és:

- Exportar una clau pública: *gpg -armor -output fileout -export ClauID*.
- Exportar una clau privada: *gpg -armor -output fileout -export-secret-key ClauID*.

L'usuària "anna" exportarà la seva clau pública i la farà arribar a l'usuari "pere", que l'afegirà al seu anell de claus. "Pere" enviarà un missatge de text xifrat usant la clau pública de l'"anna" i li enviarà per email. Ella desxifrarà el missatge utilitzant la seva pròpia clau privada.

```
# Exportar la clau pública de anna
[anna@host ~]$ gpg --armor --output anna.public.key --export 5A3946FE
[anna@host ~]$ ll
-rw-rw-r-- 1 anna anna 1772 28 gen 18:07 anna.public.key

# Enviar-la a en "pere"
[anna@host ~]$ mail -subject "clau publica anna" pere < anna.public.key

# L'usuari "pere" rep l'email i desa la clau pública de l'anna en un fitxer
[anna@host ~]$ mail
Heirloom Mail version 12.5 7/5/10. Type ? for help.
"/var/spool/mail/pere": 1 message 1 new
>N 1 anna@localhost6.local Sat Jan 28 18:16 51/2616 "ubject"
& write 1 anna.public.key
"anna.public.key" [New file] 31/1772

# Importar la clau pública de "anna" a l'anell de claus de "pere"
[pere@host ~]$ gpg --import anna.public.key
gpg: key 5A3946FE: public key "anna martinez (clau per a l'usuària anna)
     <anna@localhost.localdomain>" imported
gpg: Nombre total processat: 1
gpg: importades: 1 (RSA: 1)
```

En el següent llistat es mostra la generació del text xifrat, l'enviament de "pere" a "anna" i com es desxifra:

```
# Generar un missatge xifrat de "pere" per a "anna"
$ gpg --armor --output carta_anna.sci --recipient anna@localhost.localdomain --encrypt carta.txt
gpg: 8E452446: There is no assurance this key belongs to the named user
pub 2048R/8E452446 2012-01-28 anna martinez (clau per a l'usuària anna)
     <anna@localhost.localdomain>
     Empremta digital de la clau primària: 2091 47F7 FCA4 2EC6 69E9 7159 3CAE 98D0
     5A39 46FE
     Empremta digital de la subclau: 410B 138C 9010 4552 D4F3  FCAF 6754 E582 8E45
     2446
It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.
Use this key anyway? (y/N)y

# Enviar el missatge xifrat de "pere" a "anna"
[pere@host ~]$ mail -s "msg xifrat de pere a anna" anna < carta.xifrat.txt

# "L'anna" rep el mail, el i el desa en un fitxer
[anna@portatil ~]$ mail
Heirloom Mail version 12.5 7/5/10. Type ? for help.
"/var/spool/mail/anna": 2 messages 2 new
>N 1 anna@localhost6.local Sat Jan 28 18:16 51/2616 "ubject"
  N 2 pere@localhost6.local Sat Jan 28 19:13 34/1528 "msg xifrat de "
& write 2 carta.pere.xifrat.txt
"carta.pere.xifrat.txt" [New file] 15/722

# I ara "anna" desxifra el missatge per stdout
[anna@host ~]$ gpg --armor --decrypt carta.pere.xifrat.txt
You need a passphrase to unlock the secret key for
user: "anna martinez (clau per a l'usuària anna) <anna@localhost.localdomain>"
2048-bit RSA key, ID 8E452446, created 2012-01-28 (main key ID 5A3946FE)
Introduïu la contrasenya: privadaanna
gpg: encrypted with 2048-bit RSA key, ID 8E452446, created 2012-01-28
     "anna martinez (clau per a l'usuària anna) <anna@localhost.localdomain>"

aquest és un exemple de text pla que podria
ser un missatge.
L'ha escrit l'usuari pere i serà usat per
xifrar i per signar en altres varis exemples.
Bye
```

Per desxifrar el missatge l'usuària "anna" ha d'utilitzar la seva clau privada que es troba protegida per una contrassenya, cal escriure "privadaanna" per tal de desbloquejar-la i poder continuar. El missatge desxifrat es mostra per pantalla perquè no s'ha indicat cap opció *output*.

Igual que s'han exportat i importat claus també es poden eliminar, tant les públiques com les privades:

- `gpg --delete-key clauID` per eliminar les claus públiques.
- `gpg --delete-secret-key clau ID` per eliminar claus privades. D'una parella de claus pública/privades locals de l'usuari (no importades) primer cal eliminar la privada i posteriorment la pública.

Una clau es pot revocar per fer que deixi de tenir validesa, per exemple perquè s'ha copromés la seva seguretat, o tot i no expirar es vol fer que deixi de ser vàlida. El procediment a seguir és:

- generar un certificat de revocació per a la clau en concret: `gpg --gen-revoke clauID`.
- importar el certificat de revocació a l'anell de claus.

El sistema de claus PGP (i també la implementació GNU PG) permet crear anells de confiança. Consisteix en confiar en claus que són signades per entitats en les que es confia. Un usuari "pere" confia molt amb la seva amiga "anna" però desconeix a l'usuari "jordi". L'usuària "anna" coneix i confia amb en "jordi", tant que ha firmat ella mateixa (amb la seva clau pública) la clau pública d'en "jordi". Per dir-ho d'alguna manera "l'aval". Ara si en "pere" rep un missatge d'en "jordi" tot i no coneix l' pot confiar amb ell. Perquè? perquè ve avalat per "l'anna".

Exemple 03: Signar un contingut

Quan un emissor signa digitalment un missatge el receptor té la certesa de que el missatge ha estat enviat per l'emissor indicat i que el missatge és íntegre (no s'ha pogut modificar per altres). L'emissor ha de signar el missatge amb la seva clau privada. El destinatari pot verificar l'autenticitat utilitzant la clau pública de l'emissor.

En el següent exemple la usuària "anna" enviarà un text signat a l'usuari "pere". Per poder fer la verificació "pere" ha de disposar de la clau pública de "l'anna". O sigui que imitant el procediment descrit en l'exemple anterior caldrà exportar la clau pública de "l'anna" i que "pere" la importi al seu anell de claus.

Pedeu observar el procés d'exportació i importació de claus en l'exemple anterior.

```
# Generar un text per enviar signat
[anna@host ~]$ cat factura.txt
factura emesa per: anna
destinatari: pere
concepte:
et vaig pagar el café ahir,
em deus 1 euro!
```

```

: bye
:
: # Generar un nou fitxer signat
: [anna@host ~]$ gpg --clearsign factura.txt
: You need a passphrase to unlock the secret key for
: user: "anna martinez (clau per a l'usuària anna)"
: <anna@localhost.localdomain>
: 2048-bit RSA key, ID 5A3946FE, created 2012-01-28
: Introduïu la contrasenya: privadaanna
:
: # Mostrar el fitxer signat que s'ha generat
: [anna@portatil ~]$ cat factura.txt.asc
: -----BEGIN PGP SIGNED MESSAGE-----
: Hash: SHA1
:
: factura emesa per: anna
: destinatari: pere
: concepte:
: et vaig pagar el café ahir,
: em deus 1 euro!
: bye
: -----BEGIN PGP SIGNATURE-----
: Version: GnuPG v1.4.10 (GNU/Linux)
:
: iQEcBAEBAgAGBQJPEJFYAAoJEDyumbNa0Ub+LsoH/1VRKHpyU8XSv+NhAPG6BtPr
: qXRzHuLuhYHwNtKZj+Gtm1x30uuk6h/uLRjZYf6QA5sboLvIcYrkAa+cBquBPw3
: xY/Em9v1l1Q//NAss5L4+TV2LEwOSLzjYxNn5FFRspk9Br2BW7vT0Nex6HuaTjM2
: mz0mRyJB5NlsrBBdRNV2AVvcbjwFn5+zrd+A2DWCsgDB3AbhabeH8MNQ5B0J/4E
: +ZuIRa04DwtpubSPb5K+SIxvQMEyBL1MtW+Joth81UCIQLLK7wUIGefzZ+nFDUve
: jNK05IGLsPMG4u5+FmrGN7G3os3e8JkTa+9XXEUpaRnW1QoQjnUNsf8Wng6/dnU=
: =crIJ
: -----END PGP SIGNATURE-----
:
: # enviar el missatge al destinatari
: [anna@host ~]$ mail -s "factura de anna a pere" pere <factura.txt.asc

```

Es pot observar en el codi anterior que el fet de signar un contingut no el xifra, el contingut és en text pla. De fet es pot veure que el missatge apareix 'tal qual' i s'ha afegit al final una signatura PGP. Aquesta signatura és el *hash* del missatge codificat utilitzant la clau privada de l'usuària "anna". Per usar la clau privada cal introduir la contrassenya "privadaaana" que desbloqueja el fitxer. Finalment fixar-se que en no especificar el nom del fitxer de sortida (opció *output*) s'ha generat un fitxer de nom com el de l'origen amb l'extensió *.asc*. Un cop generat el missatge signat enviar-lo per email a l'usuari "pere".

Ara cal que l'usuari "pere" el verifiqui amb la clau pública de l'anna.

```

: # Desar el missatge rebut en un fitxer
: pere@host ~]$ mail
: Heirloom Mail version 12.5 7/5/10. Type ? for help.
: "/var/spool/mail/pere": 3 messages 1 new
:   1 anna@localhost6.local Sat Jan 28 18:16 52/2627 "ubject"
:   2 anna@localhost6.local Sat Jan 28 19:00 34/1477 "msg de anna a pere encriptat"
: >N 3 anna@localhost6.local Sat Jan 28 19:49 39/1442 "factura de anna a pere"
: & write 3 factura.anna.signada.asc
: "factura.anna.signada.asc" [New file] 20/637
:
: # Verificar el missatge
: [pere@host ~]$ gpg --verify factura.anna.signada.asc
: gpg: Signature made ds 28 gen 2012 19:41:28 CET using RSA key ID 5A3946FE
: gpg: Good signature from "anna martinez (clau per a l'usuària anna)" <anna@localhost.localdomain>
: gpg: AVIS: Aquesta clau no ve certificada per una signatura de confiança!
: gpg: No hi ha res que indique que la signatura pertany al seu propietari.
: Empremtes digital de la clau primària: 2091 47F7 FCA4 2EC6 69E9 7159 3CAE 98D0 5A39 46FE

```

Com es pot observar del codi anterior verificar el missatge és ben simple, n'hi ha prou de dir l'opció *-verify* i indicar el fitxer a verificar. El PGP identifica la signatura i busca en el seu anell de claus si disposa de la clau pública de l'emissor. Si la té la utilitza per verificar el missatge. Si no la té genera un error com el del codi següent:

```

: [pere@host ~]$ gpg --verify factura.anna.signada.asc
: gpg: Signature made ds 28 gen 2012 19:41:28 CET using RSA key ID 5A3946FE
: gpg: BAD signature from "anna martinez (clau per a l'usuària anna)" <anna@localhost.localdomain>

```

Els missatges es poden signar generant un contingut binari en lloc del contingut en text pla que genera l'opció *clearsign*. Usant l'opció *sign* es signa però es genera un fitxer binari d'extensió *.pgp*. Si s'observa el contingut del fitxer no s'entén res, es binari!. Per poder mostrar el contingut d'un fitxer signat així cal usar l'opció *decrypt*.

```

: # Signar generant un fitxer binari
: [anna@host ~]$ gpg --sign factura.txt
:
: # Ha generat un .pgp binari
: [anna@ ~]$ file factura.txt.gpg
: factura.txt.gpg: data
:
: # Observar el contingut
: [anna@host ~]$ gpg --decrypt factura.txt.gpg
: factura emesa per: anna
: destinatari: pere
: concepte:
: et vaig pagar el café ahir,
: em deus 1 euro!
:
:

```

```

: bye
: gpg: Signature made ds 28 gen 2012 20:08:26 CET using RSA key ID 5A3946FE
: gpg: Good signature from "anna martinez (clau per a l'usuària anna) <anna@localhost.localdomain>"
:
: # Extreure la part de signatura
: # gpg --detach-sign factura.txt.gpg

```

Exemple 04: Signar i xifrar un contingut

```

: # Encriptar
: [anna@host ~]$ gpg --armor --output factura.txt.asc --recipient pere --encrypt factura.txt
:
: # Mostrar el contingut encriptat (capçaleres PGP)
: [anna@host ~]$ cat factura.txt.asc
: -----BEGIN PGP MESSAGE-----
: Version: GnuPG v1.4.10 (GNU/Linux)
:
: hQEMAxLzqKP0Az8DA0gAxz4+/ChN6d/RbtMg3ZZ2cla2i9NlhNrknSaWoBkuGLaZ
: YkrG6/yggJWVCpatXsym/H/mcH0av1cej0ZaEvs0LcpuuIwsBIv0YYp5QJ5PDnu/
: bgbPz8KHKHzFmzBjrA7odg62Yjyv06q9+eaHyAmXW0b/v1VAjxT4KJ1k7L1GI1Q/
: GyfbQZuPr0unsQAFemTVZIzcabWszXFmJwnH8JgPl8BFKR9pSe1ZYCXju8AirGbJ
: XnjVUWKXAG9DH2869/SKH1iylDg0rhpzjplyQGpYGM25jE2KEIKwSj7frpdmfjN
: rirP+e+8m39fbxi80Nili/n0rKIILrf6TLJK8d0/SEtKdAwP0tZJLg9yc0R6sv50
: XMnTj1Xzg0IQ0ANryKStj8Px6Dg4y4p049Sww7dLrLCiim7k6B13Eg4t5tFA7gg1
: IAhMBQsw8/GbfMgh5Byqosmp7U23mVcL/LvNuwPiGhRJA+TynSw+TZ9om2n1WxRr
: mGFvZDjE7rjQ2qkzd60F4vJi8J3EjewFqsmL28AMNcImQLZq0Bp0uNA1yVZGcQ==
: =bMc6
: -----END PGP MESSAGE-----
:
: # Signar el missatge encriptat
: [anna@host ~]$ gpg --sign factura.txt.asc
: You need a passphrase to unlock the secret key for...
:
: # Mostrar el contingut encriptat + signat
: [anna@host ~]$ gpg --decrypt factura.txt.asc.gpg
: -----BEGIN PGP MESSAGE-----
: Version: GnuPG v1.4.10 (GNU/Linux)
:
: hQEMAxLzqKP0Az8DA0gAxz4+/ChN6d/RbtMg3ZZ2cla2i9NlhNrknSaWoBkuGLaZ
: YkrG6/yggJWVCpatXsym/H/mcH0av1cej0ZaEvs0LcpuuIwsBIv0YYp5QJ5PDnu/
: bgbPz8KHKHzFmzBjrA7odg62Yjyv06q9+eaHyAmXW0b/v1VAjxT4KJ1k7L1GI1Q/
: GyfbQZuPr0unsQAFemTVZIzcabWszXFmJwnH8JgPl8BFKR9pSe1ZYCXju8AirGbJ
: XnjVUWKXAG9DH2869/SKH1iylDg0rhpzjplyQGpYGM25jE2KEIKwSj7frpdmfjN
: rirP+e+8m39fbxi80Nili/n0rKIILrf6TLJK8d0/SEtKdAwP0tZJLg9yc0R6sv50
: XMnTj1Xzg0IQ0ANryKStj8Px6Dg4y4p049Sww7dLrLCiim7k6B13Eg4t5tFA7gg1
: IAhMBQsw8/GbfMgh5Byqosmp7U23mVcL/LvNuwPiGhRJA+TynSw+TZ9om2n1WxRr
: mGFvZDjE7rjQ2qkzd60F4vJi8J3EjewFqsmL28AMNcImQLZq0Bp0uNA1yVZGcQ==
: =bMc6
: -----END PGP MESSAGE-----
: gpg: Signature made ds 28 gen 2012 20:15:44 CET using RSA key ID 5A3946FE
: gpg: Good signature from "anna martinez (clau per a l'usuària anna) <anna@localhost.localdomain>"

```

Com es pot observar el contingut s'ha encriptat primer i signat posteriorment. Per tant la signatura digital s'ha fet generant un hash del contingut encriptat. L'usuari "pere" un cop rebí el missatge pot verificar-ne la procedència seguint el procediment habitual.

```

: # L'anna genera el missatge signat+xifrat
: [anna@host ~]$ gpg --armor --recipient pere --sign --encrypt secretsign.txt
: You need a passphrase to unlock the secret key for
: user: "anna martinez (clau per a l'usuària anna) <anna@localhost.localdomain>"
: 2048-bit RSA key, ID 5A3946FE, created 2012-01-28
:
: gpg: CE033F03: There is no assurance this key belongs to the named user
:
: pub 2048R/CE033F03 2012-01-28 pere pou prat (compte d'en pere local a la màquina)
: <pere@localhost.localdomain>
: Empremta digital de la clau primària: F881 4A48 F131 86DB F696 A02B 8D27 B681
: 3CBD FE49
: Empremta digital de la subclau: 9611 D652 F605 31CD 1DFB A9C6 1973 A8A3 CE03
: 3F03
:
: It is NOT certain that the key belongs to the person named
: in the user ID. If you *really* know what you are doing,
: you may answer the next question with yes.
: Use this key anyway? (y/N) y

```

El missatge que s'ha generat en el codi anterior s'ha signat primer i xifrat tot sencer després (també la signatura). L'usuari "pere" en rebre'l l'ha de desxifrar primer i podrà observar-ne el contingut.

```

: # Pere desxifra el missatge
: [pere@portatil ~]$ gpg --decrypt secretsign.txt.asc
:
: You need a passphrase to unlock the secret key for
: user: "pere pou prat (compte d'en pere local a la màquina)
: <pere@localhost.localdomain>"
: 2048-bit RSA key, ID CE033F03, created 2012-01-28 (main key ID 3CBDFE49)
:
: gpg: encrypted with 2048-bit RSA key, ID CE033F03, created 2012-01-28
: "pere pou prat (compte d'en pere local a la màquina) <pere@localhost.localdomain>"
: quest es un missatge que es xifra i es signa tot
: de cop, de l'anna per a en pere.
: bye
: gpg: Signature made ds 28 gen 2012 21:00:55 CET using RSA key ID 5A3946FE
: gpg: Good signature from "anna martinez (clau per a l'usuària anna) <anna@localhost.localdomain>"
: gpg: AVIS: Aquesta clau no ve certificada per una signatura de confiança!
: gpg: No hi ha res que indique que la signatura pertany al seu propietari.
: Empremtes digital de la clau primària: 2091 47F7 FCA4 2EC6 69E9 7159 3CAE 98D0 5A39 46FE

```

L'ordre dels arguments de verificació i encriptament no altera el resultat.

```
# L'anna encripta primer i després signa el missatge
[anna@host ~]$ gpg --recipient pere --encrypt --sign secretsign.txt
You need a passphrase to unlock the secret key for
user: "anna martinez (clau per a l'usuària anna) <anna@localhost.localdomain>"
# Pere desencripta i verifica el missatge
[pere@host ~]$ gpg --decrypt secretsign.txt.gpg
```

1.7.3.SMIME: Seguretat en consola.

Els missatges de correus inicialment eren exclusivament de text pla. Posteriorment va sorgir la necessitat d'incorporar altres tipus de continguts i va sorgir el format MIME. Finalment amb la proliferació d'internet amb els seus avantatges i 'inconvenients' cal un format de missatges de correu que permeti implementar propietats de seguretat. Aquest format és el format S/MIME (Secure MIME).

S/MIME és un estàndard de seguretat que utilitza criptografia de clau asimètrica (la parella de claus pública/privada) i permet xifrar i signar (autenticació, integritat i no-repudi) missatges.

Tècnicament utilitza el format PKCS#7 per al format de missatges de correu i el format X.509 per als certificats digitals.

El paquet **openssl** proporciona les prestacions de *s/mime* com una de les seves múltiples subordres. Per tant, per poder xifrar i signar missatges de correu usant S/MIME caldrà realitzar els següents passos:

- Instal·lar o disposar del software de *openssl*.
- Generar per a cada usuari un parell de claus RSA privada-pública mitjançant *openssl*.
- Utilitzar les ordres de consola *openssl smime* per realitzar l'acció que es desitgi.

Com podeu observar, el funcionament és molt similar al necessari per implementar seguretat usant GNU PG. El següent llistat de codi és un fragment extret de la pàgina de manual de *s/mime*, obtinguda amb l'ordre: **"man smime"**:

```
Create a cleartext signed message:
$ openssl smime -sign -in message.txt -text -out mail.msg -signer mycert.pem

Create an opaque signed message:
$ openssl smime -sign -in message.txt -text -out mail.msg -nodetach \
  -signer mycert.pem

Create a signed message, include some additional certificates and read the
private key from another file:
$ openssl smime -sign -in in.txt -text -out mail.msg -signer mycert.pem
  -inkey mykey.pem -certfile mycerts.pem

Create a signed message with two signers:
$ openssl smime -sign -in message.txt -text -out mail.msg \
  -signer mycert.pem -signer othercert.pem

Send a signed message under Unix directly to sendmail, including headers:
$ openssl smime -sign -in in.txt -text -signer mycert.pem \
  -from steve@openssl.org -to someone@somewhere \
  -subject "Signed message" | sendmail someone@somewhere

Verify a message and extract the signer's certificate if successful:
$ openssl smime -verify -in mail.msg -signer user.pem -out signedtext.txt

Send encrypted mail using triple DES:
$ openssl smime -encrypt -in in.txt -from steve@openssl.org \
  -to someone@somewhere -subject "Encrypted message" \
  -des3 user.pem -out mail.msg

Sign and encrypt mail:
$ openssl smime -sign -in ml.txt -signer my.pem -text | openssl smime -encrypt
  -out mail.msg \
  -from steve@openssl.org -to someone@somewhere \
  -subject "Signed and Encrypted message" -des3 user.pem

Note: the encryption command does not include the -text option
because the message being encrypted already has MIME headers.

Decrypt mail:
$ openssl smime -decrypt -in mail.msg -recip mycert.pem -inkey key.pem
```

```

The output from Netscape form signing is a PKCS#7 structure with
the detached signature format. You can use this program to verify
the signature by line wrapping the base64 encoded structure and
surrounding it with:
-----BEGIN PKCS7-----
-----END PKCS7-----
and using the command:
openssl smime -verify -inform PEM -in signature.pem -content content.txt
Alternatively you can base64 decode the signature and use:
openssl smime -verify -inform DER -in signature.der -content content.txt

Create an encrypted message using 128 bit Camellia:
$ openssl smime -encrypt -in plain.txt -camellia128 -out mail.msg cert.pem

Add a signer to an existing message:
$ openssl smime -resign -in mail.msg -signer newsign.pem -out mail2.msg

```

Creació de certificats

Primerament cada un dels usuaris que vol poder usar S/MIME ha de disposar de certificats digitals. És a dir, cal generar la parella clau pública/privada. La generació de claus privades i certificats usant *openssl* es pot consultar en altres unitats formatives d'aquest mateix mòdul: "UF2 Activitat-1 Servei Web". També es poden consultar els temes de seguretat del Mòdul 11. Finalment es poden consultar les pàgines man de *openssl* especificat la subcomanda a fer.

Per crear uns certificats per als usuaris "pere" i "anna" cal demanar a una entitat certificadora que els crei i els 'avali'. En lloc d'usar una entitat externa els pasos que es seguiran són:

1. Crear una entitat certificadora CA de nom "Veritat Absoluta".
2. Cada usuari crea la seva clau privada i la seva petició de certificat, que envia a l'entitat CA perquè sigui avalada (certificada).
3. L'entitat CA "Veritat Absoluta" firma la petició de certificat generant una clau pública o certificat per a l'usuari. Llavors li fa arribar.

S'utilitzarà un nou usuari "veritat" per fer la funció de la CA "Veritat Absoluta". Cal crear la seva clau privada i la seva clau pública autosignada.

!!

Es pot observar com generar un certificat digital de servidor SSL consultant la documentació d'aquest mòdul "UF2 Activitat-1 Servei web".

```

[veritat@host ~]$ openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for ca.key: cakey

[veritat@host ~]$ openssl req -new -x509 -nodes -sha1 -days 365 -key ca.key -out ca.crt
Enter pass phrase for ca.key: cakey
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:ca
State or Province Name (full name) []:catalunya
Locality Name (eg, city) [Default City]:barcelona
Organization Name (eg, company) [Default Company Ltd]:VeritatAbsoluta
Organizational Unit Name (eg, section) []:certificats
Common Name (eg, your name or your server's hostname) []:veritat
Email Address []:

```

Cada un dels usuaris "anna" i "pere" ha de generar la seva clau privada i la petició de certificat.

```

[anna@host certs]$ openssl genrsa -des3 -out anna.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for anna.key: annakey
Verifying - Enter pass phrase for anna.key: annakey

[anna@host certs]$ openssl req -new -key anna.key -out anna.csr
Enter pass phrase for anna.key: annakey
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:ca
State or Province Name (full name) []:catalunya
Locality Name (eg, city) [Default City]:barcelona
Organization Name (eg, company) [Default Company Ltd]:personal
Organizational Unit Name (eg, section) []:personal
Common Name (eg, your name or your server's hostname) []:anna

```



```
Email Address []:anna@localhost.localdomain

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:anna
An optional company name []:anna

[anna@host certs]$ cp anna.csr /tmp/m08/
```

L'entitat CA “Veritat Absoluta” rep la petició i genera un certificat per a l'usuari. Per fer-ho utilitza un fitxer de configuració on cal especificar clarament que el certificat ha de permetre validar “**emailProtection**”. Un cop generat cal fer-lo arribar a l'usuari.

```
[veritat@host ~]$ cat ca.conf
basicConstraints = critical,CA:true
extendedKeyUsage = serverAuth,emailProtection

[veritat@host ~]$ openssl x509 -CA ca.crt -CAkey ca.key -req -in /tmp/m08/anna.csr \
-days 365 -sha1 -extfile ca.conf -CAcreateserial -out anna.crt
Signature ok
subject=/C=ca/ST=catalunya/L=barcelona/O=personal/OU=personal/CN=anna
/emailAddress=anna@localhost.localdomain
Getting CA Private Key
Enter pass phrase for ca.key: cakey

[veritat@host ~]$ cp anna.crt /tmp/m08/
```

Un cop generats els certificats apropiats per a cada usuari ja poden usar les ordres de consola de S/MIME per intercanviar continguts xifrats i signats de manera similar a com es feia amb el GNU PG.

1.7.4.Seguretat en clients gràfics: thunderbird.

Com enviar correu segur utilitzant clients de correu gràfics? Els clients de correu actuals permeten incorporar tots els mecanismes de seguretat que s'han estat tractant: xifrat, autenticació, integritat i no repudi. Poden portar incorporades aquestes característiques o es poden configurar per afegir-les.

Els següents exemples treballen amb el client gràfic *thunderbird*. Per poder implementar OpenPGP i S/MIME al *thunderbird* cal instal·lar els paquets corresponents a *enigmail*. En reiniciar l'aplicació apareixeran nous menús i botons amb les opcions de seguretat.

```
[root@host ~]# rpm -qa | grep enigmail
thunderbird-enigmail-1.0.1-1.fc12.i686
```

La Figura 19 “Thunderbird amb enigmail instal·lat” mostra la pantalla de *thunderbird* on es poden observar les opcions i botons que apareixen de nou.

Figura 19. Thunderbird amb enigmail instal·lat.



De la figura anterior es pot observar l'aparició de:

- Un nou menú general anomenat “open PGP” que permet configurar: 'les preferències', 'la gestió de claus' i cridar 'l'assistent del PGP'.

- En la finestra de redactar un nou botó “Open PGP” permet seleccionar si el missatge s’ha de xifrar i/o signar.
- En la finestra de redactar un nou botó “S/MIME” permet també xifrar i/o signar el missatge i observar les propietats de seguretat.
- En la part inferior dreta de la redacció de missatges apareixen dues ‘icones’ petites noves que permeten seleccionar si el missatge ha d’anar signat i/o xifrat. De fet n’hi ha dos pel Open PG i dos més pel S/MIME que només apareixen si s’hi han configurat els certificats digitals.

Per provar el funcionament es pot generar un compte de correu al *thunderbird* per amb un dels usuaris locals usats (“pere” o “anna”) o també es pot crear un compte de correu extern de “gmail” per exemple. La Figura 20 “Resum inicial de detecció d’un compte local” mostra el resum de dades efectuat en la detecció automàtica del thunderbird per a l’usuari local “pere”.

Figura 20. Resum inicial de detecció d’un compte local.

The screenshot shows the 'Paràmetres del compte' (Account Parameters) window in Thunderbird. The left sidebar lists various settings for two accounts: 'pere@localhost.localdomain' and 'ecanet1profe@gmail.com'. The 'pere@localhost.localdomain' account is selected, and its 'Paràmetres del servidor' (Server Parameters) are displayed in the main pane. The settings include:

- Tipus de servidor:** Servidor de correu IMAP
- Nom del servidor:** imap.googlemail.com
- Port:** 993
- Per defecte:** 993
- Nom d'usuari:** ecanet1profe@gmail.com
- Paràmetres de seguretat:**
 - Connexió segura: TLS sobre SSL
 - ☐ Utilitza l'autenticació segura
- Paràmetres del servidor:**
 - ☒ En iniciar, comprova si hi ha correu nou
 - ☒ Comprova si hi ha nous missatges cada 10 minuts
 - Quan suprimeixi un missatge:
 - ☒ Mou-lo a aquesta carpeta: Paperera
 - ☐ Marca'l com a suprimir
 - ☐ Suprimeix-lo immediatament
 - ☐ Neteja ("Esborra") la Safata d'entrada en sortir
 - ☐ Buida la paperera en sortir
- Director local:** /home/pere/.thunderbird/q1zlcrrg.default/ImapMail/imap.googlemail.com

At the bottom, there are buttons for 'Accions del compte', 'Avançat...', 'Cancel·la', and 'D'acord'.

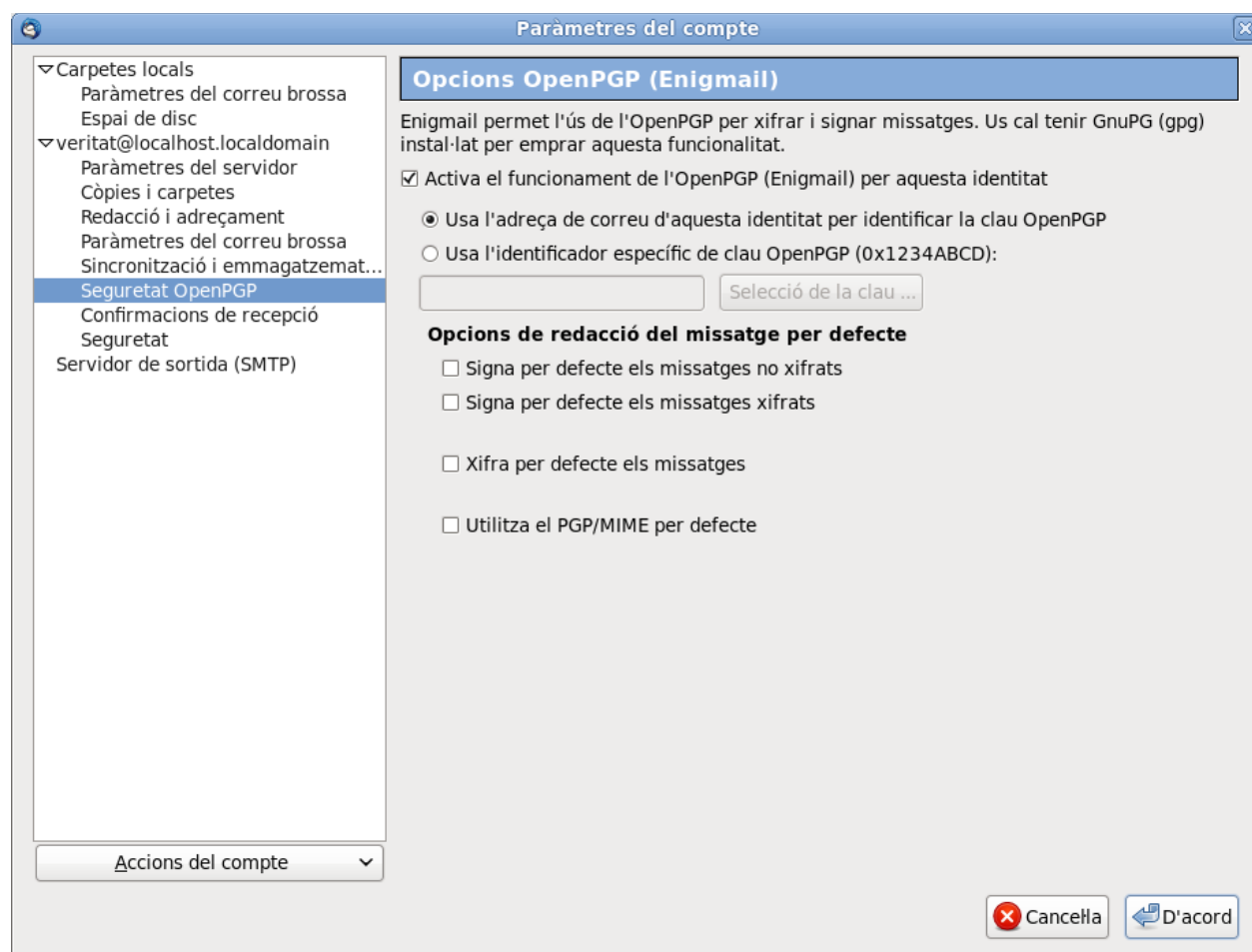
Per un usuari donat es poden crear múltiples comptes dins del thunderbird. Un exemple interessant de fer és provar un compte extern d'un webmail que en permeti l'accés via POP o IMAP, com per exemple *Gmail*. Es poden consultar els passos per configurar aquests tipus de comptes a la documentació “help” de *Gmail*. Però segurament no serà necessari perquè el pròpi *thunderbird* detectarà la configuració automàticament tant bon punt s'indiqui un nom de compte d'aquest domini. La Figura 21 “Resum inicial de detecció d'un compte de Gmail” mostra el procés de creació/detecció d'un compte a *Gmail*.

Figura 21. Resum inicial de detecció d'un compte de Gmail.



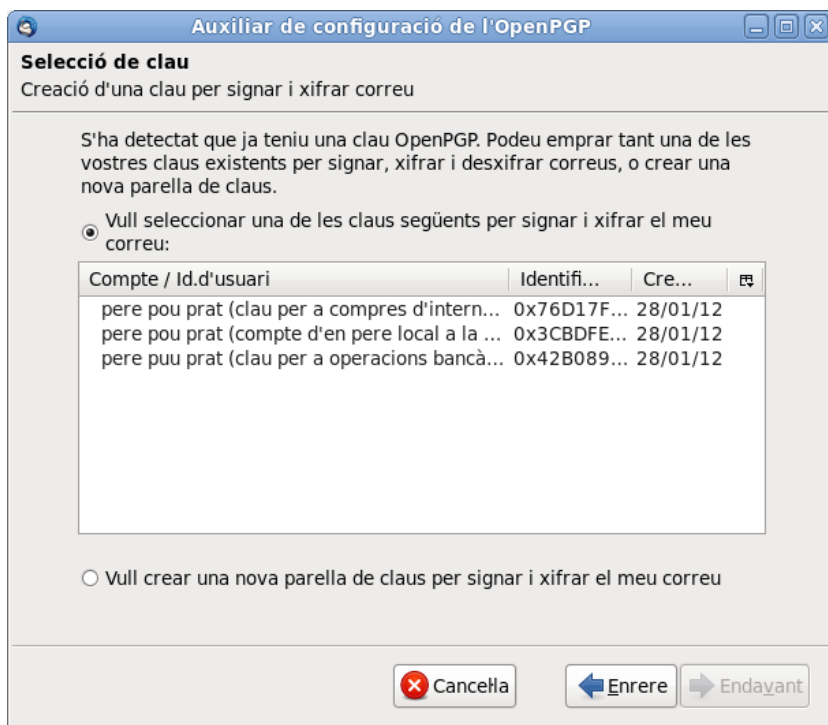
Si l'usuari del que s'està creant el compte de correu en thunderbird ja disposa de claus PGP (o GNU PG com és el cas), es detecten automàticament i s'incorporen a la configuració. Si l'usuari disposa de més d'una clau s'ecull quina cal usar. La Figura 22 "Configuració de 'Seguretat Open PGP' dels paràmetres d'un compte" mostra els detalls referents a la configuració de Open PGP de la configuració d'un compte de correu.

Figura 22. Configuració de "Seguretat Open PGP" dels paràmetres d'un compte.



La Figura 23 "Pantalla de selecció de clau a usar per un compte" mostra que l'usuari "pere" pot escollir quina de les claus vol usar associada al seu compte local.

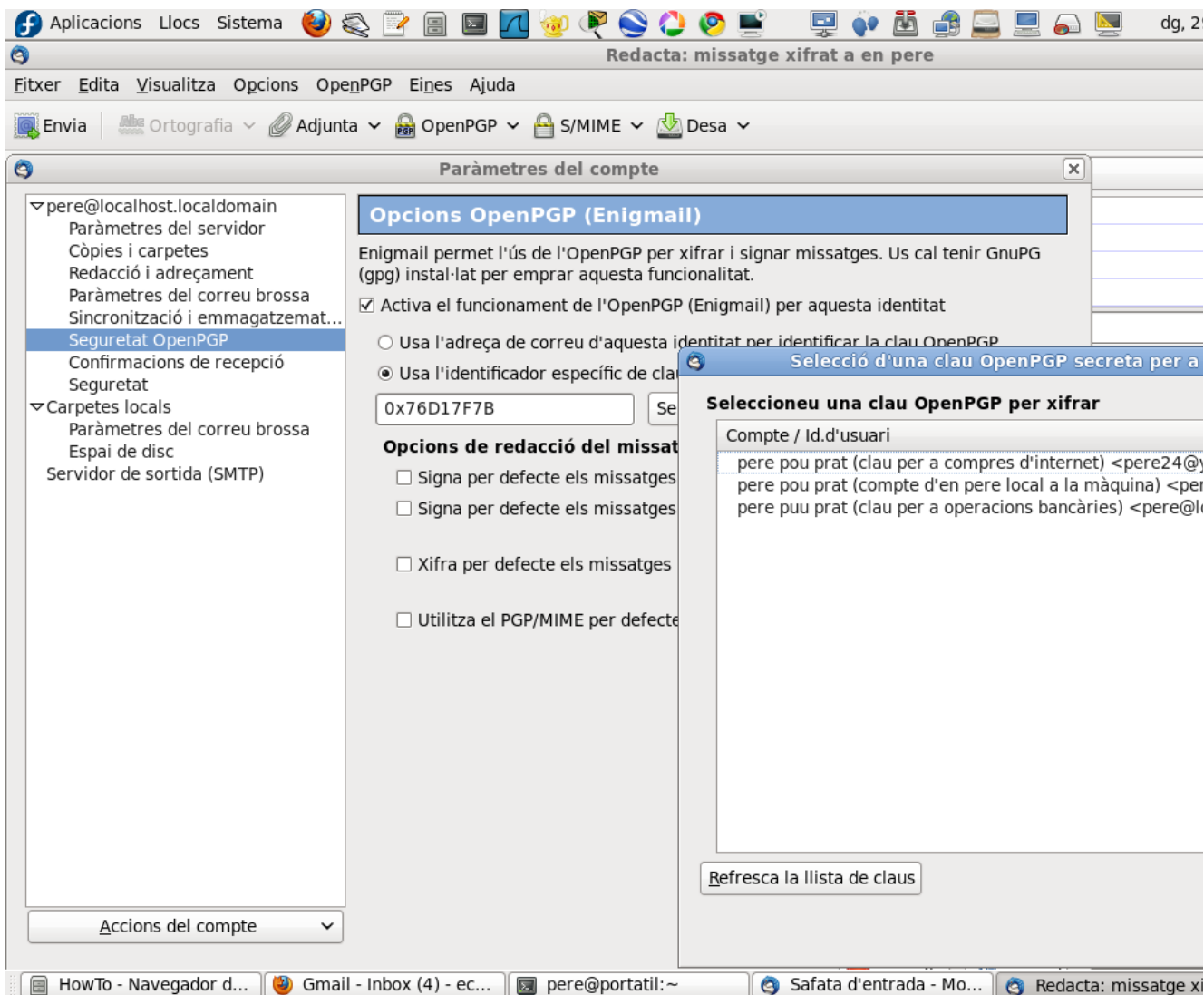
Figura 23. Pantalla de selecció de clau a usar per un compte.



Si no es disposa de claus generades anteriorment, si no s'han detectat apropiadament o si simplement s'en volen crear unes de noves, el menú de OpenPGP permet iniciar "l'auxiliar de configuració". De fet en la figura anterior "Figura 23" es pot observar l'opció que permet "crear una nova parella de claus".

En la Figura 24 "Auxiliar de creació de noves claus" es pot observar la pantalla de l'auxiliar per a la creació de noves claus per a un usuari de thunderbird.

Figura 24. Auxiliar de creació de noves claus.



La Figura 25 es pot observar que a l'usuari "pere" se li ha afegit una nova clau generada des del thunderbird per l'auxiliar de configuració. Observar també que en el llistat de claus apareix la clau pública de l'usuària anna, que s'ha importat manualment. De fet es pot observar que l'usuari "pere" disposa de quatre parelles de claus públiques/privades, mentre que de l'usuària "anna" hi ha la clau pública importada però evidentment no hi ha la clau privada (només ella la té). Clicant en qualsevol de les claus es mostren les dades detallades de la clau seleccionada.

Figura 25. Llistat de claus de l'usuari "pere".

Auxiliar de configuració de l'OpenPGP

Creació de clau

Creació d'una clau per signar i xifrar correu

Heu de crear una 'parella de claus' per signar i xifrar correu, o per llegir correus xifrats. Una parella de claus té dues claus, una pública i una privada.

Us cal donar la vostra clau pública a qualsevol de la vostra llista de contactes que desitgi verificar la vostra signatura, o per xifrar el correu que us envii. Mentrestant heu de mantenir en secret la vostra clau privada. Mai l'heu d'entregar o deixar-la sense protecció. Permet llegir tot el correu xifrat que la gent us envii. També permet xifrar correu en el vostre nom. Com que és secreta, està protegida per una contrasenya.

Compte / Id.d'usuari:
veritat <veritat@localhost.localdomain> - veritat@localhost.localdomain

Contrasenya

Confirmeu la vostra contrasenya teclejant-la una altra vegada

A les tres claus existents s'ha afegit la nova clau creada des de l'auxiliar de configuració.

En resum, per poder generar correu xifrat i signat usant *thunderbird* aquestes són accions que cal fer:

- Instal·lar el software de *enigmail*.
- Si no es disposa de comptes de correu crear tants comptes de correu com es desitgin. Un usuari del sistema pot tenir diferents correus. Per exemple un compte local, dos comptes a *Gmail* de finalitats diferents, un altre a *Yahoo*, un altre compte no webmail en un el servidor de l'empresa (o més d'un!) i tants altres com vulgui o el 'transtorn de personalitat múltiple' li permeti.
- Quan es generen comptes de correu nous, *thunderbird* intenta detectar automàticament els paràmetres de configuració de correu sortint SMTP i correu entrant IMAP o POP.
- (GPG) Si l'usuari disposa de claus PGP (o GNU PG) s'intentaran carregar per defecte. Si no en té es suggerirà la creació d'un joc de claus.
- (S/MIME) si l'usuari disposa de certificats digitals pròpis incorporar-los i configurar el panell se 'seguretat'. En qualsevol cas sempre es poden generar noves claus de de "l'auxiliar de configuració" o des de la gestió de claus "Genera noves claus".
- Establir les opcions per defecte referents a les accions de signar i xifrar.

Un cop un usuari disposa de claus instal·lades ja pot enviar correu signat i xifrat? Doncs si i no. Anemem a veure el perquè:

- **signar**: per signar correu l'usuari utilitza la seva clau privada, per tant pot enviar correu signat a qualsevol destinatari.
- **xifrar**: per xifrar correu es requereix la clau pública del destinatari. Per tant, només podrà enviar correu als usuaris que li hagin proporcionat la seva clau pública. Si "pere" vol enviar correu xifrat a "anna" abans ha d'haver aconseguit la clau pública de la usuària "anna".
- **propagar claus**: d'alguna manera els usuaris s'han de fer arribar entre ells les claus públiques. Existeixen vàries mecanismes com per exemple la centralització en servidors de claus. Els usuaris que volen donar a conèixer la seva clau pública a la resta de la comunitat publiquen les seves claus en servidors de domini públic. Els clients de correu com *thunderbird* tenen pre configurades vàries adreces de servidors públics de claus.

De fet un mecanisme més simple (potser fins i tot més 'intim') entre els usuaris és enviar-se les claus els uns als altres. Es pot fer de diferents formes però la més pràctica és que quan un usuari envia un missatge i hi adjunta automàticament la seva clau pública. Així qualsevol destinatari li pot respondre de manera confidencial si així ho vol. D'aquesta manera

els usuaris s'aprenen les claus dels uns als altres. Aquest és el cas més típic en xarxes locals i entre col·legues.

1.8. Configurat el servidor de correu com a servei segur.

Els protocols de correu analitzats en aquesta documentació són tots protocols de transport d'informació en text pla. Així qualsevol comunicació SMTP, POP o IMAP es realitza en text pla i pot ser 'monitoritzada' per tercers que tinguin accés als nodes de xarxa per on passen aquestes comunicacions. De fet s'han vist exemples de monitorització de les converses TCP utilitzant l'eina *wireshark*. De fet no és únic dels protocols de correu sinó que aquesta feblesa és comuna a la majoria de protocols d'internet com també li passa a HTTP, FTP, TFTP, etc.

En els protocols de correu existeixen mecanismes similars als usats en HTML per poder establir canals de comunicació segurs:

- SMTPS
- POPS
- IMAPS

SMTPS No és un protocol nou i diferent de SMTP ni una extensió seva, és simplement una manera d'anomenar al fet d'utilitzar SMTP per sobre de SSL o TLS. Igual que l'usuari està acostumat a comunicacions HTTP segures usant HTTPS pot usar transport de correu segur amb SMTPS. El protocol és el mateix però viatja per sobre de SSL que li proporciona la seguretat. Per usar SMTPS cal comunicar-se amb un port diferent del 25, és el port 465. Això representa un problema ja que el sistema de correu a internet es basa majoritàriament en l'utilització del port 25, aquest problema es va resoldre amb l'utilització de STARTTLS.

POPS Igual que passa amb HTTPS i SMTPS POPS no és de per si un protocol sinó simplement un acrònim per descriure l'utilització de l'accés remot a bústies POP usant un transport segur SSL o TLS. Utilitza el port 995 en lloc del port 110 clàssic de POP.

IMAPS Es també la manera d'anomenar el protocol IMAP viatjant sobre una capa de transport segur com SSL o TLS. El port utilitzat per IMAP sobre SSL és el port 993.

Els acrònims SMTPS, POPS i IMAPS indiquen l'utilització del protocol usant una capa de transport segura SSL o TLS, permetent una comunicació xifrada que no pot ser monitoritzada per tercers (igual que passa amb HTTP sobre SSL que s'anomena HTTPS).

Per poder usar SSL o TLS amb algun dels protocols de correu cal que el servidor i/o el client disposin de certificats digitals apropiats per certificar la funció que fan (de servidor o de client). De la mateixa manera que per poder implementar una web segura amb HTTPS usant SSL cal un certificat de servidor SSL per poder implementar un servidor SMTPS segur caldrà un certificat de servidor per emetre el correu i un certificat de client per rebre correu. El mateix succeeix amb els protocols POPS i IMAPS, per poder usar SSL requereixen de certificats digitals.

El següent llistat de codi mostra els continguts del paquet de software *uw-imap* usat per treballar com a servidor POP i IMAP usant tant accés amb text pla com amb accés segur:

```
root@host ~]# rpm -ql uw-imap
/etc/pam.d/imap
/etc/pam.d/pop
/etc/pki/tls/certs/imapd.pem
/etc/pki/tls/certs/ipop3d.pem
/etc/xinetd.d/imap
/etc/xinetd.d/imapd
/etc/xinetd.d/imapd
/etc/xinetd.d/ipop2
/etc/xinetd.d/ipop3
/etc/xinetd.d/pop3s
/usr/sbin/imapd
/usr/sbin/ipop2d
/usr/sbin/ipop3d
/usr/share/doc/uw-imap-2007e
/usr/share/doc/uw-imap-2007e/SSLBUILD
/usr/share/man/man8/imapd.8uw.gz
/usr/share/man/man8/ipopd.8uw.gz
```

En altres apartats d'aquesta documentació s'ha instal·lat i configurat el servidor *uw-imap* per usar-lo com a servidor POP i IMAP.

Del llistat anterior es pot observar que el servidor incorpora dos

certificats digitals anomenats *imapd.pem* i *ipop3d.pem*. Aquests són els certificats que li permeten implementar SSL.

```
/etc/pki/tls/certs/imapd.pem
/etc/pki/tls/certs/ipop3d.pem
```

El següent llistat de codi mostra que els certificats són vàlids per a fer de servidor i de client SSL:

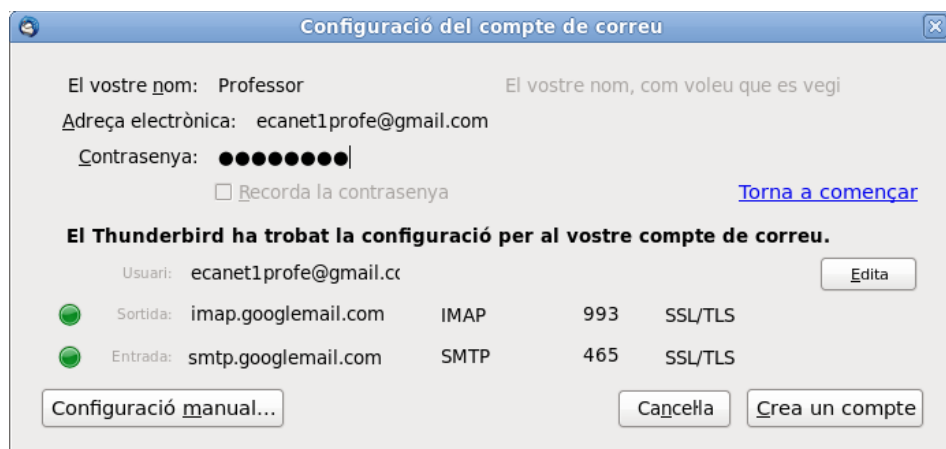
```
root@host ~]# openssl x509 -purpose -noout -in /etc/pki/tls/certs/imapd.pem
Certificate purposes:
SSL client : Yes
SSL client CA : Yes
SSL server : Yes
SSL server CA : Yes
Netscape SSL server : Yes
Netscape SSL server CA : Yes
S/MIME signing : Yes
S/MIME signing CA : Yes
S/MIME encryption : Yes
S/MIME encryption CA : Yes
CRL signing : Yes
CRL signing CA : Yes
Any Purpose : Yes
Any Purpose CA : Yes
OCSP helper : Yes
OCSP helper CA : Yes
Time Stamp signing : No
Time Stamp signing CA : Yes
```

Usant l'utilitat *nmap* es pot observar els ports segurs per on escolta el servidor les comunicacions POP3 (995) i IMAPS (993):

```
[root@host ~]# nmap localhost
Starting Nmap 5.21 ( http://nmap.org ) at 2012-02-22 17:34 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000020s latency).
rDNS record for 127.0.0.1: localhost.localdomain
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
110/tcp   open  pop3
111/tcp   open  rpcbind
143/tcp   open  imap
631/tcp   open  ipp
993/tcp   open  imaps
995/tcp   open  pop3s
```

En la Figura 26 “Configuració de compte de correu a Gmail” es pot veure la pantalla de creació d'un compte de correu en el *thunderbird* que configura l'accés remot a un compte de correu de *Gmail*. Es pot observar que el correu entrant utilitza IMAP sobre SSL/TLS al port 993 del servidor *imap.googlemail.com* que és com s'anomena el servidor de correu de *Gmail*. També es configura com a correu sortint el servidor SMTP sobre SSL/TLS de google. S'utilitza el port 465 del servidor *smtp.googlemail.com* que és el nom del host de *Gmail* que fa la funció de servidor SMTP.

Figura 26. Configuració de compte de correu a Gmail.



Per els atrevits a parlar SSL el següent exemple mostra com iniciar un diàleg en mode consola amb un servidor SSL. S'utilitza una de les utilitats de l'ordre *openssl* que permet actuar com un client SSL, en aquest exemple contacta amb el servidor *imap* de *Gmail*:

```
[root@host ~]# openssl s_client -crlf -connect imap.gmail.com:993
... output suprimit ...
```


En el següent llistat es connecta amb el servidor de correu SMTP de *Gmail* al port segur 465 usant SSL:

STARTTLS

El fet d'utilitzar un port diferent per a les comunicacions SMTP segures del port SMTP usual representava un problema per a les comunicacions de correu a internet ja que la majoria de serveis de correu estan configurats per usar el port 25. Un solució que va sorgir a aquest problema és la capacitat de poder iniciar comunicacions en SMTP pla al port 25 i convertir-les en comunicacions segures TLS.

Aquesta mena de 'truc de màgia' permet iniciar un diàleg TCP al port 25 d'un servidor SMTP i detectar si es permet o no l'establiment de comunicacions segures. Si no es permet es pot realitzar igualment la comunicació amb SMTP pla. Si es permet la connexió segura es pot canviar al mode segur i la comunicació SMTP passara a viatjar per sobre d'una capa TLS segura. L'avantatge d'aquest mecanisme és que fa innecessària la utilització d'un altre port i la necessitat de que el servidor estigui escoltant per ports diferents.

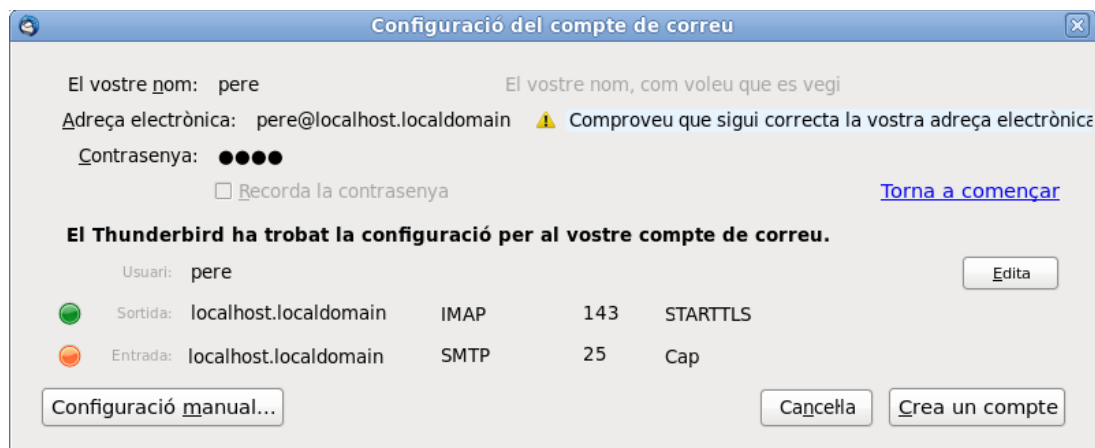
El mecanisme **STARTTLS** permet que comunicacions TCP iniciades de forma insegura o en text pla per els ports usals es converteixin en comunicacions segures que viatgen usant la capa de transport SSL o TLS.

El mecanisme STARTTLS no és únic del correu SMTP sinó que també es pot utilitzar per als protocols POP, IMAP, XMPP (jabber o el protocol per xatejar), NNTP (les news) i LDAP. De fet és una extensió que tenen aquests protocols de text pla per convertir-se en protocols segurs.

El funcionament és ben senzill, el client es connecta al servidor i realitza la comanda **STARTTLS**, es produeix la negociació SSL apropiada i si té èxit tot el que es realitza a partir d'aquest moment ja és usant una comunicació segura. Per poder usar per sota TLS o SSL evidentment cal que el servidor disposi dels certificats digitals apropiats i de la configuració apropiada que li permet oferir la capacitat STARTTLS als clients. El mateix haurà de fer el client en aquells casos en què també es requereixi la identificació del client.

En la Figura 27 "Configuració de compte de correu local." s'ha configurat un compte de correu local al *thunderbird*. S'utilitza com a servidor de correu entrant el servidor IMAP al port de text pla 143, però la comunicació es una comunicació segura perquè permet l'utilització de STARTTLS (per això surt el semàfor verd). En canvi la configuració del correu sortint s'ha fet amb el servidor *sendmail* local que no s'ha configurat per permetre connexions segures ni tampoc per permetre STARTTLS (per això el semàfor es taronja, funciona però en mode insegur).

Figura 27. Configuració de compte de correu local.



El següent llistat mostra l'establiment d'una sessió en text pla al servidor local IMAP (el servidor és el *uw-imap*) al port 143. El servidor respon mostrant quines són les seves característiques o capacitats i entre elles s'observa que proporciona la prestació STARTTLS:

```
[root@host ~]# telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4REV1 I18NLEVEL=1 LITERAL+ SASL-IR LOGIN-REFERRALS STARTTLS]
  localhost.localdomain IMAP4rev1 2007e.404 at Wed, 22 Feb 2012 18:08:06 +0100 (CET)
001 starttls
001 OK STARTTLS completed
```

En canvi en el següent llistat es mostra que el servidor local *sendmail* no permet la connexió segura usant STARTTLS. En iniciar una sessió *telnet* al port 25 del servidor i intentar iniciar STARTTLS s'observa que falla:

```
[root@host ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 localhost6.localdomain6 ESMTP Sendmail 8.14.4/8.14.4; Wed, 22 Feb 2012 18:14:09 +0100
HELP
214-2.0.0 This is sendmail
214-2.0.0 Topics:
214-2.0.0      HELO      EHLO      MAIL      RCPT      DATA
214-2.0.0      RSET      NOOP      QUIT      HELP      VRFY
214-2.0.0      EXPN      VERB      ETRN      DSN       AUTH
214-2.0.0      STARTTLS
214-2.0.0 For more info use "HELP <topic>".
214-2.0.0 To report bugs in the implementation see
214-2.0.0      http://www.sendmail.org/email-addresses.html
214-2.0.0 For local information send email to Postmaster at your site.
214 2.0.0 End of HELP info
STARTTLS
454 4.3.3 TLS not available after start
```

En un servidor *sendmail* configurat apropiadament per usar STARTTLS s'observaria:

```
[root@host ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 localhost6.localdomain6 ESMTP Sendmail 8.14.4/8.14.4; Wed, 22 Feb 2012 18:14:09 +0100

STARTTLS
220 2.0.0 Ready to start TLS
```

Túnel SSH

Un mecanisme simple se seguretat usat en molts entorns corporatius és utilitzar túnel ssh. Qualsevol comunicació entre dos hosts que utilitzi un protocol insegur basat en text pla es pot fer passar per un túnel SSH de manera que el protocol insegur viatga dins d'una connexió SSH que com sabem són comunicacions xifrades segures.

Està clar que és més estable configurar apropiadament els servidors de correu per treballar amb SSL i STARTTLS però el mecanisme dels túnel SSH és simple i fàcil d'implementar i s'utilitza per resoldre casos puntuals de comunicació entre dos hosts. Imaginem per exemple que tot el tràfic de correu del host "departament" va redireccionat per regles del *firewall* al host "servidor". Aquest tràfic extrem a extrem es pot posar dins un túnel SSH en lloc de preocupar-se de fer la configuració usant SSL.

Tot tràfic basat en text pla entre dos hosts (tràfic de tipus extrem a extrem) es pot fer passar dins d'un túnel SSH de manera que passa a ser tràfic segur.

Per implementar tràfic usant túnel ssh es pot consultar l'extensíssima documentació de Open SSH i la documentació del mòdul de seguretat de ASIX M11 "Seguretat i alta disponibilitat". El següent fragment és part del "**man ssh**" on es mostra un exemple de túnel ssh d'una sessió de xat IRC:

```
The following example tunnels an IRC session from client machine "127.0.0.1" (localhost)
to remote server "server.example.com":
$ ssh -f -L 1234:localhost:6667 server.example.com sleep 10
$ irc -c '#users' -p 1234 pinky 127.0.0.1
```

Sendmail amb SSL

A continuació es descriu el procés necessari per configurar *sendmail* per poder usar STARTTLS. Per usar STARTTLS i SSL/TLS cal que *sendmail* disposi de certificats digitals de servidor i de client. Cal configurar *sendmail* indicant-li que permeti usar comunicacions segures i indicar on es troba cada un dels certificats. Un cop reiniciat el servei ja es possible establir comunicacions segures.

El següent llistat mostra com generar automatitzadament els certificats de *sendmail* utilitzant l'utilitat *make* de *openssl*:

```
# Crear els certificats digitals
[root@host certs]# pwd
/etc/pki/tls/certs

[root@host certs]# make sendmail.pem
umask 77 ; \
    PEM1="/bin/mktemp /tmp/openssl.XXXXXX" ; \
    PEM2="/bin/mktemp /tmp/openssl.XXXXXX" ; \
    /usr/bin/openssl req -utf8 -newkey rsa:2048 -keyout $PEM1 -nodes -x509 -days 365 -out $PEM2
    -set_serial 0 ; \
    cat $PEM1 > sendmail.pem ; \
    echo "" >> sendmail.pem ; \
    cat $PEM2 >> sendmail.pem ; \
    rm -f $PEM1 $PEM2
Generating a 2048 bit RSA private key
.....+++
writing new private key to '/tmp/openssl.tl53ul'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:ca
State or Province Name (full name) []:bcn
Locality Name (eg, city) [Default City]:bcn
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:sendmail local
Email Address []:
```

Per configurar *sendmail* es fa en dos passos, en el primer s'edita el fitxer de configuració *sendmail.mc* que utilitza *macros m4* i és més fàcil de configurar (això diuen!). Un cop establertes les opcions apropiades cal generar el vertader fitxer de configuració, el fitxer *sendmail.cf*. Per fer-ho cal disposar de les utilitats que proporciona el paquet *sendmail-cf*.

```
# Identificar el fitxer de configuració (de macros)
[root@phost certs]# locate sendmail.mc
/etc/mail/sendmail.mc

# Editar les línies que es mostren a continuació
[root@phost certs]# vim /etc/mail/sendmail.mc
define(`confAUTH_OPTIONS', `A p')dnl

TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl

define(`confCACERT_PATH', `/etc/pki/tls/certs')dnl
define(`confCACERT', `/etc/pki/tls/certs/ca-bundle.crt')dnl
define(`confSERVER_CERT', `/etc/pki/tls/certs/sendmail.pem')dnl
define(`confSERVER_KEY', `/etc/pki/tls/certs/sendmail.pem')dnl
```

El següent llistat genera el vertader fitxer de configuració i reinicialitza el servei. Després s'executa l'opció de *sendmail* que en llista les propietats i es pot observar que permet usar STARTTLS.

```
# Instal·lar l'utilitat m4 per a sebdmail
[root@host certs]# yum install sendmail-cf

# Generar el nou fitxer de configuració .cf
[root@host certs]# m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

# Reiniciar sendmail
[root@portatil certs]# service sendmail restart
S'està aturant el sm-client: [ FET ]
S'està aturant el sendmail: [ FET ]
S'està iniciant el sendmail: [ FET ]
S'està iniciant el sm-client: [ FET ]

# Observar que permet STARTTLS
[root@portatil certs]# sendmail -d0.1 -bv
Version 8.14.4
Compiled with: DNSMAP HESIOD HES_GETMAILHOST LDAPMAP LOG MAP_REGEX
MATCHGECOS MILTER MIME7TO8 MIME8TO7 NAMED_BIND NETINET NETINET6
NETUNIX NEWDB NIS PIPELINING SASLv2 SCANF SOCKETMAP STARTTLS
TCPWRAPPERS USERDB USE_LDAP_INIT

===== SYSTEM IDENTITY (after readcf) =====
(short domain name) $w = localhost6
(canonical domain name) $j = localhost6.localdomain6
(subdomain name) $m = localdomain6
(node name) $k = portatil
=====
```

Un cop està tot configurat apropiadament els clients ja es poden connectar al servidor SMTP al port 25 usant text pla i passar a iniciar una sessió segura amb SSL usant STARTTLS.

```
[root@host certs]# telnet localhost 25
Trying 127.0.0.1...
```

```

Connected to localhost.
Escape character is '^]'.
220 localhost6.localdomain6 ESMTP Sendmail 8.14.4/8.14.4; Wed, 22 Feb 2012 19:50:33 +0100
STARTTLS
220 2.0.0 Ready to start TLS

```

La Figura 28 “Configuració de compte de correu local amb STARTTLS” mostra el panell resum de *thunderbird* on un usuari anomenat “usuari” configura el servidor de correu sortint al servidor *sendmail* local acabat de configurar. Es pot observar que permet l'utilització de STARTTLS connectant al port 25 (el port usual).

Figura 28. Configuració de compte de correu local amb STARTTLS.



1.9. Elabora documentació relativa a la instal·lació, configuració i recomanacions d'utilització del servei.

L'administrador del servei de correu ha d'enregistrar clarament la configuració global del servidor SMTP i dels serveis POP i/o IMAP que s'utilitzin.

Caldrà descriure:

- Quins serveis de correu s'han configurat, per a quins ports i quin tipus de seguretat empren.
- Llistat de l'estructura de directoris (tipus tree) usats pel servidor.
- Llistat dels directoris de configuració SSL/TLS o STARTTLS (claus i certificats).
- Llistat de l'estructura utilitzada per a realitzar certificats digitals. Si s'utilitza una CA pròpia anotar les dades descriptives de l'entitat.
- Anotar la configuració de sendmail utilitzada, els canvis a la configuració estàndard que s'han fet i documentar la finalitat de cada línia de configuració modificada, afegida o eliminada.