Apunts sobre el correu electrònic

Vocabulari i software:

*MTA (Mail Transfer Agent): Software servidor que s´encarrega d´enviar els correus a altres servidors MTA (o bé rebre'ls d'altres servidors MTA). Fan servir el protocol SMTP. Exemples de software servidor MTA són:
Postfix (http://www.postfix.org)
Exim (http://www.exim.org)
Sendmail (http://www.sendmail.org)
CourierMTA (http://www.courier-mta.org)

Una altra tasca que avui dia també integren tots els programes anteriors és la de MDA (Mail Delivery Agent) -antigament ho feien programes separats-, la qual consisteix bàsicament en recollir tots els missatges rebuts pel servidor MTA final (provinents d'altres servidors MTA) i col.locar-los a la bústia personal de l'usuari de destí. Altres tasques d'un software MDA és el filtratge, l'ordenació o la resposta automàtica de correus (quan el destinatari està de vacances,per exemple).

*Servidor POP: Software que permet als usuaris registrats accedir a la seva bústia personal on trobarà els missatges allà presents (prèviament col.locats allí per algun software MDA). Fan servir el protocol POP3. Opcionalment podrà descarregar aquests missatges al seu ordinador local. Un servidor POP és senzill d´operar i no necessita massa potència...normalment vénen empaquetats juntament amb els servidors IMAP (veure següent paràgraf).

*Servidor IMAP: Software que permet als usuaris registrats accedir a la seva bústia personal on trobarà els missatges allà presents (prèviament col.locats allí per algun software MDA). Fan servir el protocol IMAP4. A diferència dels servidors POP, però, no es poden descarregar els missatges a l'ordinador local de l'usuari. Una altra de les diferències entre el protocol POP i IMAP és que el primer no permet reorganitzar la bústia en diferents nivells de subcarpetes i/o etiquetes però el segon sí. Un servidor IMAP requereix força RAM i molt d´espai d´emmagatzematge. Exemples de software servidor POP/IMAP són:
Dovecot (http://www.dovecot.org)
CourierIMAP (http://www.courier-mta.org/imap)
CyrusIMAP (http://cyrusimap.web.cmu.edu)
UW Imap Toolkit (http://www.washington.edu/imap)

*MUA (Mail User Agent) : Software que utilitza l´usuari per escriure i enviar nous missatges (fent servir per això un determinat servidor MTA que el MUA ha de tenir configurat) i també per visualitzar els missatges que hi hagi a la seva bústia personal (havent-se de connectar el MUA, per fer això, amb un determinat servidor IMAP/POP que tingui configurat). Exemples de software MUA:
Thunderbird (http://www.mozilla.org)
Evolution (http://projects.gnome.org/evolution)
Mutt (http://www.mutt.org) -Per consola-
Alpine (http://www.washington.edu/alpine) -Per consola també-

A més d'un servidor MTA-MDA, un servidor POP/IMAP i un client MUA, que seria el mínim necessari per implementar un sistema complet d'enviament i rebuda de correus, es pot afegir a més altre software addicional:

*Eliminadors d'spam i/o detectors de virus als missatges (s'instal.len en els servidors MTA-MDA):
SpamAssassin (http://spamassassin.apache.org) -AntiSpam-
Assp (http://assp.sourceforge.net) -AntiSpam-
MailScanner (http://www.mailscanner.info) -AntiSpam i AntiVirus-
Amavis-new (http://www.ijs.si/software/amavisd) -AntiVirus-
ClamAV (http://www.clamav.net) -AntiVirus-

*Gestors de llistes de correu (s'instal.len en els servidors MTA-MDA):
Majordomo (http://www.greatcircle.com/majordomo)
MailMan (http://www.list.org)

*Clients WebMail: Software MUA que funciona sobre un servidor web, de manera que l'usuari no ha de tenir instal.lat localment cap MUA sinò que amb un navegador pot accedir, com si fos una pàgina web més, a la seva bústia per llegir

els missatges rebuts i enviar-ne d'altres. Generalment aquest tipus de software està escrit amb un llenguatge d´script de servidor com PHP o similar. Exemples de software WebMail són:
SquirrelMail (http://www.squirrelmail.org)
Horde IMP (http://www.horde.org/imp)
Ilohamail (http://ilohamail.org)
Sqwebmail (http://www.courier-mta.org/sqwebmail)
RoundCube (http://roundcube.net)
Rainloop (http://www.rainloop.net)
Webmail-Lite (https://afterlogic.org/webmail-lite)

*Suites completes integrades (inclouen servidor MTA-MDA, servidor IMAP/POP, client WebMail, etc):
Mail-in-a-box (https://mailinabox.email)
Les següents suites a més contenen agendes i calendaris, capacitats de missatgeria instantània, etc, de manera que són més aviat una "suite de col.laboració".
Zimbra (http://www.zimbra.com)
Zarafa (http://www.zarafa.com)
Kolab (https://kolab.org)
Citadel (http://www.citadel.org)

Autenticació mitjançant SASL i xifratge mitjançant TLS:

Des de sempre als servidors POP o IMAP ha existit l'autenticació per poder accedir a la bústia personal particular de cada usuari loguejat. No obstant, per fer servir un servidor SMTP (és a dir, MTA) antigament no calia registrar-se: es podien enviar correus sense qualsevol tipus de control. Això, òbviament, va originar el problema de l'spam, així que avui dia tots els servidors SMTP comercials implementen un sistema d'accés on s'ha de tenir un compte d'usuari registrat per fer-lo servir com a remitent però si instal.lem un servidor SMTP propi, cal ser conscients de què no hi ha cap sistema d'accés implementat per defecte.

Avui dia la majoria de servidors SMTP (i també d'altres tipus com els IMAP, XMPP o LDAP) estan compilats amb una llibreria anomenada SASL ("Simple Authentication and Security Layer"), la qual permet posar d'acord diferents servidors SMTP per tal de consensuar el mecanisme d'autenticació que faran servir entre ells (i poder-se enviar, per tant, correus entre ells sense rebutjar-los). Si no es fa servir SASL, caldria especificar a mà el mètode d'autenticació a usar en cada servidor SMTP per on el missatge de correu hagués de passar. Cal tenir present, en aquest sentit, que per defecte, un servidor SMTP només accepta enviar correus a un altre servidor SMTP remot si aquests correus estan originats en clients de la seva pròpia xarxa però no pas si provenen d'altres servidors SMTP externs: és per això que un mecanisme d'autenticació oferit per SASL i acordat entre les parts de forma automàtica és vital. Mecanismes d'autenticació oferits per SASL molt comuns són PLAIN, LOGIN, DIGEST-MD5 o GSSAPI, entre d'altres (ho veurem més endavant).

D'altra banda, antigament tampoc s'usava cap mètode d'encriptació per enviar (i consultar) els correus, de manera que els missatges viatjaven per la xarxa totalment en obert. Avui dia es fa servir el protocol TLS per oferir els xifratge dels missatges. Com que aquest protocol se situa, dins la pila TCP/IP, entre la capa de transport i la capa d´aplicació, en aquest cas estarem parlant dels protocols SMTPS POPS o IMAPS. La seva implantació però, és opcional, així que ens podem trobar servidors comercials (pocs) que no ofereixin l'encriptació de missatges.

El protocol TLS permet també realitzar autenticacions (i per tant, permet no haver s'usar SASL) però per diversos motius històrics avui dia els servidors SMTP usen una combinació de les dues tecnologies: SASL per autenticar i TLS per xifrar. En aquest cas, un cop autenticat l'usuari via SASL, per començar l'enviament de correus encriptats amb TLS cal que s'executi la comanda SMTP especial "STARTTLS", la qual indica el "pas" de comunicació no xifrada a comunicació xifrada.

Degut a totes aquestes variants (amb o sense autenticació, amb o sense xifratge), al llarg de la història els protocols SMTP, POP i IMAP han anat ampliant-se, de manera que avui dia a la pràctica ens podem trobar amb les següents casuístiques:

Servidor SMTP (sense autenticació o amb autenticació SASL només): escolta al port 25
Servidor SMTP (amb autenticació SASL + STARTTLS): escolta al port 587
Servidor SMTP (amb autenticació TLS i xifratge TLS) : escolta al port 465 . Actualment en desús
Servidor POP (amb autenticació SASL només): escolta al port 110
Servidor POP (amb autenticació TLS i xifratge TLS) : escolta al port 995
Servidor IMAP (amb autenticació SASL només) : escolta al port 143
Servidor IMAP (amb autenticació TLS i xifratge TLS) : escolta al port 993

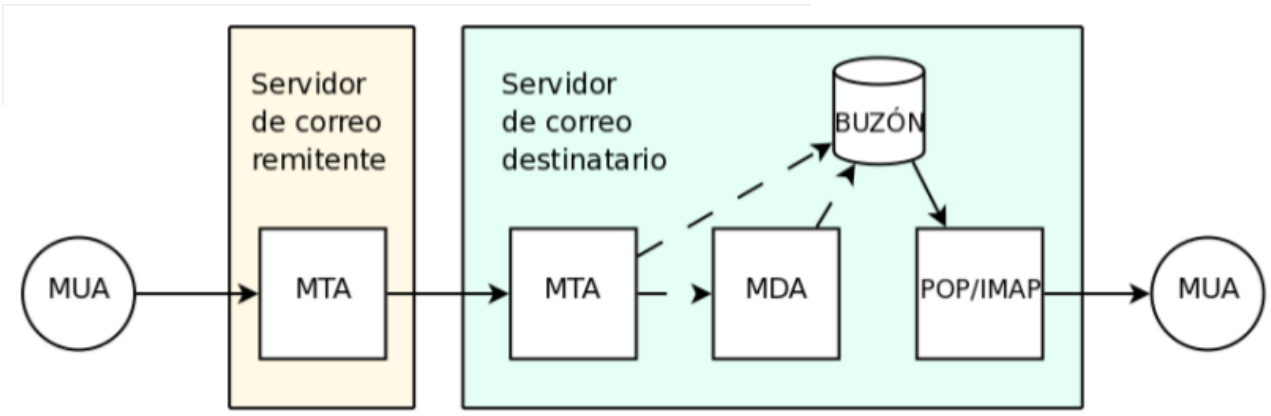
Esquema dels passos realitzats en l´enviament i rebuda d´un missatge de correu:

Básicamente, el envio de emails es un procedimiento donde se envia texto de una máquina a otra, añadiéndose al buzón de correo personal del destinatario. Este buzón, por cierto, actualmente puede estar en formato "mbox" o bien "Maildir" (la diferencia básica entre uno y otro es que el primero implica que todos los mensajes del buzón se encuentran dentro de un único fichero de texto mientras que el segundo implica que cada mensaje será guardado en un fichero de texto por separado).

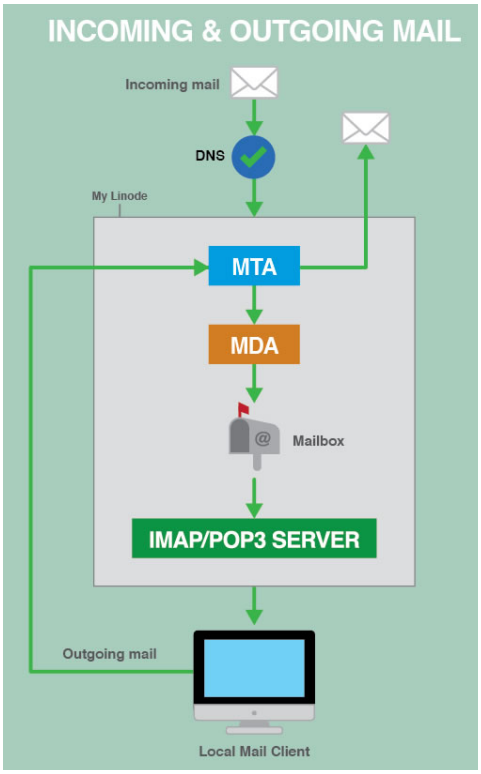
No obstante, este procedimiento es largo y complejo porque intervienen varios programas trabajando en cadena (incluyendo un servidor DNS, imprescindible para que un servidor SMTP sepa quien és el servidor SMTP siguiente!) para conseguir que el email llegue a su destino. Veamos en detalle cómo funciona el correo electrónico. Ann (*ann@domain.com*) quiere enviar un email a Bob (*bob@otherdomain.com*). Gráficamente:

Protocol SMTP
MUA (usuari *ann@domain.com*) ----------> Servidor SMTP configurat al MUA d'Ann (màquina *mail.domain.com*)
|
| Consulta prèvia al valor del registre MX del
| servidor DNS de "otherdomain.com" per
| saber qui és *mail.otherdomain.com*
|
V Protocol SMTP
|
| El missatge arriba al MTA de
| *mail.otherdomain.com* i d'aquí passa
| al MDA, el qual el col.loca
| a la bústia de l'usuari *bob*
Protocol POP/IMAP V
MUA (usuari *bob@otherdomain.com*) <------- Servidor POP3/IMAP configurat al MUA de Bob (normalment és també la máquina *mail.otherdomain.com* però podria ser una altra màquina)

O més bonic (encara que aquí no es mostra el pas intermig entre el quadrat groc i el verd, que és la consulta que ha de fer el servidor de correu groc a algun servidor DNS disponible per saber quin és el valor del registre MX associat al domini de destí -és a dir, la IP del servidor de correu verd-):



Un altre esquema similar és el següent:



Tal como se puede ver en los diagramas anteriores, el email llegará a su destino siguiendo estos pasos:

1.-Ann utiliza algún software cliente de correo (un MUA:como Outlook, Evolution, Thunderbird, Mutt ...) para crear el correo, y enviarlo al servidor SMTP que dicho MUA tenga configurado en primera instancia (en nuestro ejemplo, *mail.domain.com*). Este servidor SMTP normalmente primero comprovará que Ann pueda hacer uso de él comprobando que exista en él un usuario llamado "ann" (es decir, autentićandola); si es así, el servidor SMTP recoge el correo de Ann y lo pone en la cola (en el caso de Postfix, por ejemplo, la cola es una simple carpeta: /var/spool/postfix).

2.-El servidor SMTP de Ann, *mail.domain.com*, enviará el correo al servidor SMTP correspondiente al dominio que se halla especificado como destinatario (es decir, si el destinatario es *b@otherdomain.com*, se buscará el servidor SMTP correspondiente al dominio *otherdomain.com*, y se le enviará el correo a él). Este proceso puede no ser directo, y se puede enviar el correo a través de varios SMTP hasta llegar al correcto. ¿Cómo sabe la máquina *mail.domain.com* cuál es el servidor SMTP de *otherdomain.com*? Pues realizando una consulta Dns del registro MX para ese dominio de destino!

3.-Una vez el correo ha llegado al servidor SMTP de destino, éste lo puede enviar a algún agente intermediario MDA que lo haga pasar por algún escaneador de spam o algún antivirus. Finalmente, el correo se repartirá en el buzón adecuado (del usuario bob, en este caso). Tal como hemos dicho, existen básicamente dos formatos de buzón: Mbox y Maildir. Se recomienda emplear el segundo, ya que es más moderno y menos propenso a errores; en el formato Maildir un buzón es en realidad una carpeta Maildir dentro de la carpeta Home del usuario en cuestión donde se almacenarán los diferentes mensajes, cada uno en un fichero diferente (a diferencia del formato Mbox, que guarda todos los mensajes dentro de la carpeta /var/spool/mail/usuario en un único fichero).

4.-El correo ya está almacenado en el buzón, pero para acceder a él (esta vez comunicándose mediante el protocolo POP/IMAP contra el software servidor correspondiente, que no es el mismo que el software servidor SMTP), Bob tendrá que utilizar algún software cliente de correo (un MUA, otra vez). Evidentemente, para acceder al buzón previamente se tendrá que autenticar como un usuario válido en el servidor POP/IMAP.En principio, los usuarios del servidor de correo son usuarios reales de la máquina (aunque se puede configurar para que se extraigan de una base de datos tipo MySQL, o mejor, de un servidor LDAP, por ejemplo). Si accede mediante POP3, tendrá la posibilidad de descargar físicamente los mensajes desde su buzón en el servidor a una carpeta local.

Capçaleres d´un missatge de correu:

Received:	Aquesta capçalera apareixerà per cadascun dels servidors SMTP que intervinguin en l´enviament del missatge (com si fos un "matasegells" del correu convencional). L´ordre de les diferents capçaleres Received que apareguin ve determinat per l´ordre de rebuda del missatge, de manera que la capçalera de més a dalt serà la corresponent a la del servidor SMTP final, i la capçalera de més abaix correspondrà al primer servidor SMTP (el que tindrà configurat el remitent). És a dir, la traça del camí del missatge s´ha de llegir des de la capçalera Received de més assota fins a la de més a sobre. Un valor típic d´aquesta capçalera seria aquest: *from servidorSMTPanteriorqueestaalacapçaleraReceiveddesota (nomdns [ipseva])* *by servidorSMTPactual (ipseva)* *with ESMTP (SoftwareServidor)* *via nosequeesaixo* *id identificadorMissatge* *for usuari@destinatari* *; data hora +0200* on la data i hora són quan el servidor Smtp en qüestió ha rebut el missatge, i es prenen sempre amb l´horari GMT. Per saber la data i hora local del servidor Smtp en qüestió, s´ha de saber que el número positiu del final indica el número d´hores i minuts a l´est del meridià de Greenwitch, i si fos un número negatiu indicaria el número d´hores a l´oest del meridià de Greenwitch. Això vol dir que si es vol saber l´hora local, si el número és positiu s´ha de restar al temsp GMT, i si és negatiu, s´ha de sumar.
Message-ID:	Identificador del missatge que assigna el primer servidor SMTP (el que té configurat el remitent). Aquest identificador és únic per cada mail enviat per aquest servidor d´origen.
From:	Direcció del remitent. Normalment agafa el valor que s´escrigui a "mail from:" (direcció que obligatòriament ha de ser vàlida pel servidor Smtp si aquest demana autenticació, i corresponent a l´usuari autenticat), però es pot manipular manualment dins el missatge per a què valgui qualsevol cosa. L´efecte de posar un valor en aquesta capçalera diferent de "mail from:" serà que el destinatari veurà (en primera instància) que la direcció d´origen no és la real.
Return-Path: Reply-To:	Direcció a la què s´enviaran les respostes del missatge actual. Agafarà sempre el valor del que s´escrigui a "mail from: ", encara que la capçalera From: es manipuli manualment al missatge.
To:	Direcció del destinatari. Normalment agafa el valor que s´escrigui a "rcpt to:", però es pot manipular manualment dins el missatge per a què valgui qualsevol cosa. L´efecte de posar un valor en aquesta capçalera diferent de "rcpt to:" serà que el destinatari de "rcpt to:" rebrà el correu però veurà com que la direcció de destí no és la seva.
Cc:	Llista de direccions (separades per comes) a les que també s´enviarà una còpia del missatge
Bcc:	Similar a Cc, amb la diferència que cada destinatari es veurà com a únic destinatari existent (és a dir, no sabrà a qui més s´ha enviat el missatge, cosa que amb Cc es pot veure)
Date:	Data i hora a la que es va enviar el missatge
Delivery-Date:	Data i hora a la que es va dipositar el missatge al servidor Pop/Imap final
Subject:	Texte que va escriure el remitent a la línia d´"assumpte"
MIME versió:	Versió de l´estàndar MIME (Multipart Internet Mail Extension), mitjançant el qual es pot enviar mails en jocs de caràcters diferents de l´Ascii, en formats diferent del texte pla -per exemple, en Html-, i també arxius adjunts.
Content-Type:	Originalment els correus només contenien texte Ascii. Per afegir la possibilitat de poder utilitzar altres tipus de jocs de caracters a l´hora de redactar els missatges (i així incloure caràcters no anglosaxons, per exemple), i també per utilitzar diferents formats als missatges (per incloure codi Html, per exemple), i també per incloure arxius adjunts, es va desenvolupar l´estàndar MIME. Aquest estàndar perment que el remitent indiqui al destinatari quin tipus de missatge rebrà,informant-lo del joc de caracters emprat, el format utilitzat, els arxius adjunts presents... per a què pugui interpretar-ho correctament. En concret, si és el cas de què s´incorpora algun arxiu adjunt, el que es fa és transformar els bits de l´arxiu (binari, de música, una foto...) en una codificació textual en format base64 i enviar aquesta, avisant al destinatari del tipus de contingut que s´ha codificat, per a què ell la pugui decodificar. Un valor possible podria ser: *text/plain;charset="iso-8059-1"* on s´especifica que el missatge és texte pla i a més està codificat amb el joc de caràcters iso-8059-1, que és un dels jocs que conté les lletres accentuades, la ñ, la ç, etc Altres valors podrien ser image/png , application/pdf , multipart/x-mime(per indicar adjunts...)
Content-Location:	El seu valor és la ruta on es troba originalment el fitxer adjunt que s´enviarà codificat dins del cos del missatge.
Content-Description:	Opcional. Breu descripció sobre el format, codificació o adjunts del missatge

Content-Transfer-Encoding:	Pot valer ascii, base64, ...
Content-ID:	Identificador únic d'una part del cos del missatge en un missatge partit amb contingut “multipart”
In-Reply-To:	El seu valor és el message-id d´un altre missatge respecte el qual està relacionat d´alguna manera
References:	(bé perquè el contesta, o en fa referència)
Comments:	Evident.
Keywords:	

Totes les capçaleres que comencen per X són opcionals. Les més comunes són:

X-Mailer:	Programa o sistema de correu que va utilitzar el remitent per enviar el correu
X-Sender:	El mateix que From:, però més difícil de falsificar perquè és la direcció que va anotar el servidor
Sender:	de correu del remitent i a penes pot ser manipulada. En realitat, en teoria From: indica quin usuari crea el missatge, i Sender: indica qui l´envia.
X-Authenticated-IP:	Informa de la IP de la xarxa interna de l´ordinador que va emetre el missatge. Figurarà si l´equip des del que es va enviar el missatge forma part d´una xarxa i si els serveis de correu de la mateixa estan així configurats
X-Originating-IP:	Informa de la IP de la màquina original que ha generat el missatge. En principi, aquesta informació es pot obtenir de la primera capçalera Received, però no està de més tenir una confirmació extra.
X-Priority:	Prioritat del missatge: alta, normal, etc
X-MSPriority:	

Per veure més informació, consultar els RFCs o bé http://www.activexperts.com/activemail/headers

Protocol POP:

Les respostes del servidor només són de dos tipus: *+OK missatge* ó *-ERR missatge*

USER usuari	Evident
PASS contrasenya	Evident
QUIT	Evident
LIST [nºmissatge]	Mostra una llista numerada dels missatges que hi ha a la bústia, i el número de bytes que ocupa cada missatge. La numeració està ordenada de missatge més antic a més modern. Si s´especifica un missatge concret, es dóna la mateixa informació però només per aquest missatge.
STAT	Mostra el número de missatges totals que hi ha a la bústia, i el número de bytes totals que ocupa el conjunt de tots els missatges
RETR nºmissatge	Obté el contingut (incloent les capçaleres) del missatge especificat.
TOP nºmiss nºlin	Obté el contingut (incloent les capçaleres) del missatge especificat, però només fins la línia especificada amb el número que és el segon paràmetre, de tal manera que només es veuran les línies que hi hagin fins allà. Cada capçalera conta com una línia. Un efecte interessant d´aquesta comanda és que NO marca com a llegits els missatges, cosa que sí que fa RETR.
DELE nºmissatge	Marca com eliminat el missatge especificat. Però atenció, el missatge només s´elimina de forma efectiva quan es tanqui sessió amb QUIT.
RSET [nºmissatge]	Anula tots els DELE que s´hagin realitzat dins la sessió actual, desmarcant-los de ser esborrats. Si s´especifica un missatge concret, es fa el mateix però només per aquest missatge.
NOOP	Com diu el seu nom, no fa res (NO OPeration). Serveix simplement per indicar al servidor que es vol mantenir la conexió viva, ja que normalment els servidors tenen un temps de timeout després del qual, si no hi ha activitat, tanquen la conexió.

Altres comandes POP són: APOP, AUTH, UIDL...

NOTA: Els RFCs del protocol POP3 es troben llistats a http://en.wikipedia.org/wiki/Post_Office_Protocol#Related_Requests_For_Comments_.28RFCs.29

Protocol IMAP:

Tant POP com IMAP són protocols d´accés a la bústia personal de correu, però tenen diferències de funcionament:

> *POP permet descarregar els missatges en forma de fitxers a bústies locals, per tal de poder-los llegir de forma off-line (amb la possibilitat d´eliminar-los a la vegada del servidor, o no). En canvi, amb IMAP sempre accedirem al nostre correu a través de la bústia remota del servidor. Això té l´inconvenient de que haurem d´estar permanentment connectats als servidor per poder treballar dins la nostra bústia, però té la ventatge de poder consultar el correu des de qualsevol punt client.
>
> *IMAP és un protocol molt més complet i ric: aporta la possibilitat de crear/eliminar/renombrar subbústies dins la bústia principal, permet establir diferents flags als missatges ("nou", "llegit", "eliminat",...) i realitzar cerques de missatges segons determinats criteris (el seu contingut, la data d´emisió o rebut, el tamany, als arxius adjunts només, etc). També permet l´accés simultani des de vàries màquines a una mateixa bústia, cosa que Pop no ho permet.

Alguna de les comandes més importants del protocol IMAP són les següents (no hi són pas totes!). Per cert, si es volen provar -amb *ncat*, per exemple-, hem de tenir present que davant de qualsevol comanda haurem d´escriure un parell de lletres qualssevol. Això és perquè el servidor ens respondrà amb el missatge adient precedit de les mateixes lletres que s´han escrit, identificant així que és la resposta a una consulta determinada. Per exemple, si la comanda fos CHECK, s´hauria d´escriure "ab CHECK" , on "ab" són dues lletres qualssevol.

També s´ha de tenir present que sempre existirà una bústia anomenada "INBOX", a partir de la qual es poden crear les que es desitgin.

Les respostes del servidor, depenent de la comanda, poden ser: *OK missatge* ó *NO missatge* (quan no es permet realitzar la consulta) ó *BAD missatge* (quan la consulta està mal formulada o no es reconeix)

CAPABILITY	Mostra les capacitats del servidor. És el missatge que mostra el servidor quan es conecta.
STARTTLS	Habilita el xifratge de la transacció
LOGIN usuari contrasenya	Evident. Vigilar d´executar aquesta comanda després de STARTTLS, perquè si no les dades del login poden ser esnifades
LOGOUT	Tanca la connexió amb el servidor
SELECT nombustia	Sel.lecciona una bústia per tal de poder-hi executar comandes concretes, com ara EXPUNGE, CLOSE, SEARCH, FETCH, STORE ó COPY. A més, el servidor retornarà una resposta amb dades interessants: *FLAGS* (Indica els flags presents a la bústia: \Answered, \Deleted, \Seen, \Draft, \ NoSelect, \Flagged, etc *nº EXISTS* (indica el número de missatges que hi ha a la bústia) *nº RECENT* (indica quins d´aquests estan sense llegir) *OK UNSEEN* nº (indica el número del primer correu sense llegir) *READ-WRITE* | *READ-ONLY* (indica l´estat de la bústia: en el cas de la comanda SELECT, serà READ-WRITE sempre)
EXAMINE nombustia	Similar a SELECT, però la bústia estarà en mode READ-ONLY
STATUS nombustia	Alternativa a EXAMINE per obtenir dades una bústia, la qual ja ha sigut sel.leccionada prèviament. Així d´evita haver de desel.leccionar la bústia activa També retorna MESSAGES, RECENT I UNSEEN. Es pot obtenir només un d'aquests elements si s'indica *STATUS nombustia (RECENT)*, per exemple (en aquest cas, per obtenir només els missatges recents.
CREATE nombustia CREATE nomsubbustia	Crea una bústia (que pot estar a l´interior d´una altra, la qual ja ha d´existir prèviament)
DELETE nombustia DELETE nomsubbustia	Elimina una bústia (que pot estar a l´interior d´una altra). Si es vol eliminar una bústia que conté altres a dins, no es podrà: s´ha de començar eliminant les més internes.
RENAME nomantic nomnou	Canvia el nom de la bústia especificada, pel nou nom. Es pot canviar el nom també a la bústia "INBOX", però en aquest cas, la bústia "INBOX" continuarà existint, buida.

LIST referencia nombustia	L´ús més típic és fer *LIST “” “*”*, per veure el contingut de totes les bústies, indicant si tenen subbústies i els seus flags. El caràcter “*” significa qualsevol caràcter (inclós el caràcter separador de bústies -per defecte, “/”). També es podria fer servir el caràcter “%”, que és similar però sense incloure aquest caràcter “/”.
EXPUNGE	Esborra definitivament els missatges marcats com \Deleted de la bústia actual
CLOSE	Esborra definitivament els missatges marcats com \Deleted i surt de la bústia actual
SEARCH *criteri1 criteri2 NOT criteri3*	Busca els missatges de la bústia actual que compleixin el/s criteri/s especificat/s. Si s´especifica més d´un, s´ha d´entendre com un AND lògic. Alguns dels criteris poden ser: ALL ANSWERED ó DELETED ó DRAFT ó FLAGGED ó SEEN ó RECENT UNANSWERE ó UNDELETED ó UNDRAFT ó UNFLAGGED ó UNSEEN NEW (similar a RECENT UNSEEN) ó OLD (similar a NOT RECENT) FROM cadena ó TO cadena BCC cadena CC cadena BEFORE data ó SINCE data ó ON data SENTBEFORE data ó SENTSINCE data ó SENTON data BODY cadena ó HEADER capçalera valor ó SUBJECT cadena TEXT cadena (seria similar a buscar en el body i en el head) LARGER nºbytes ó SMALLER nºbytes
FETCH ...	Aquesta comanda és força complexa. Permet veure el contingut dels missatges, canviar-li flags, copiar-lo a una altre bústia...Aquí només posaré uns quants exemples: *Per recuperar els flags de tots els missatges de la bústia: *fetch 1:* (UID FLAGS)* retornarà una cosa semblant a ** 1 FETCH (UID 1 FLAGS (\Recent \Draft))* Amb aquesta comanda recuperem un llistat de tots els missatges (en aquest cas, només 1) en els quals se´ns indiquen els flags i l´identificador per poder-nos referir a cada missatge amb un número que l´identifica de forma unívoca. El primer paràmetre és un rang (1:* és del primer fins l´últim, podríem posar 2:3 per recuperar el missatge 2 i 3 o simplement 4 si volem recuperar només el 4) *Per recuperar l´estructura d´un missatge concret amb: *fetch 4 (BODYSTRUCTURE)* que ens retornarà un “xoriço” de dades: *1 FETCH (BODYSTRUCTURE ((”TEXT” “HTML” (”CHARSET” “ISO-8859-1") NIL NIL “7BIT” 155 8 NIL NIL NIL NIL)(”APPLICATION” “OCTET-STREAM” (”NAME” “prueba.TMP”) NIL NIL “BASE64" 63146 NIL (”ATTACHMENT” (”FILENAME” “prueba.TMP”)) NIL NIL) “MIXED” (”BOUNDARY” “———— 080306020500030305000606") NIL NIL NIL))* *8 OK FETCH completed* Aquí podem veure que tenim un arxiu adjunt al missatge de nom ‘prueba.tmp’ *(”APPLICATION” “OCTET-STREAM” (”NAME” “prueba.TMP”) NIL NIL “BASE64" 63146 NIL (”ATTACHMENT” (”FILENAME” “prueba.TMP”))* *Si volem recuperar parts d´un missatge concret podem fer: *fetch 1 (BODY[0])* recupera les capçaleres. També funcionaria *fetch 1 (BODY[HEADER])* *fetch 1 (BODY[1])* recupera el cos. També funcionaris *fetch 1 (BODY[TEXT])* *fetch 1 (BODY[2])* recupera el primer adjunt
NOOP	No fa res. Simplement manté viva la comunicació amb el servidor per a què aquest no la doni per tancada

NOTA:Es pot veure també http://www.cicei.com/ocon/gsi/tut_tcpip/3376imap.html per més informació

NOTA: Els RFCs del protocol IMAP es troben llistats a http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol#External_links

Protocol SMTP:

La redacció i enviament de correus utilitzant directament el protocol SMTP requereix la realització d´una sèrie de passos. Suposarem en aquest primer exemple que ja estem connectats al servidor SMTP (amb ncat o similar com per exemple així: *ncat smtp.gmail.com 25*), i que aquest servidor no demana cap tipus d´autenticació per fer ús dels seus serveis. Aquesta configuració no és realista avui dia (seria una bomba d´spam), però és el cas més senzill:

1.-El primer pas és saludar al servidor. En la majoria dels casos, si no es fa aquest pas previ, el servidor no ens deixarà fer res. Això és degut a què com que els servidors SMTP poden rebre conexions tant de clients directes com d´altres servidors SMTP que li envien correu, abans de fer qualsevol cosa ha d´haver una salutació prèvia. Per saludar al servidor es poden utilitzar dues comandes:

HELO nommaquinaclient (Serveix per presentar-se, dient quina màquina s´és, i per rebre un missatge de benvinguda del servidor. Es pot observar com a la resposta del servidor se´ns informa de que coneix la quina és la nostra ip. En comptes de posar el nom de la nostra màquina, de fet, es podria posar qualsevol cosa.)

EHLO nommaquinaclient (Versió extesa de HELO. Si el servidor entén aquesta comanda, vol dir que implementa les extensions del protocol (RFC1651). A més de rebre el mateix missatge de benvinguda, el servidor ens respondra a aquesta comanda, mostrant a més una llista concreta de quines d´aquestes extensions entén, com per exemple els tipus d´autenticació que soporta (AUTH PLAIN, AUTH LOGIN, etc), si admet xifratge de la comunicació (STARTTLS), quin és el tamany màxim admés per un missatge de correu -en bytes- (SIZE nºbytesmaximmissatge), si admet rebre del client una seqüència de comandes sense haver de respondre a cadascun per separat (PIPELINING), etc.

2.-Escriure la direcció del remitent, així (les majúscules no importen, però els dos punts sí!):

MAIL FROM: direccio@remitent.com

Si el servidor SMTP no tingués activat cap mètode d'autenticació, aquí es podria escriure qualsevol direcció fictícia. Però si si el té, només acceptarà un usuari existent

NOTA: Hi ha servidors que obliguen a escriure la direcció del remitent entre < i >.

3.-Escriure la direcció de destí (les majúscules no importen, però els dos punts sí!):

RCPT TO: direccio@desti.org , unaaltra@desti.net

4.-Notificar que començarem a escriure el missatge:

DATA

5.-Opcionalment, podem escriure capçaleres amb el valor que volguem, una per línia amb el format nom:valor

Subject: Hola

6.-Començar a escriure el missatge pròpiament dit. Si s´han escrit capçaleres, S´HA DE DEIXAR UNA LÍNIA EN BLANC DE SEPARACIÓ ENTRE LES CAPÇALERES I EL COS DEL MISSATGE.

7.-Per acabar, s´ha d´escriure en una línia sola un punt. És a dir, la marca de final de missatge és <SaltLinia>Punt<SaltLinia>

8.-Es pot continuar enviant missatges, o bé, desconnectar-se del servidor SMTP, amb la comanda QUIT.

És a dir, seria una cosa similar a això:

ehlo qualsevolcosa
mail from: correuremitent@dominiServidorSmtp
rcpt to: correudestinatari@dominiServidorSmtp
data
<Capçaleres a enviar, si es vol>

<El texte a enviar>
.
quit

Les respostes del servidor SMTP consten d´un codi numèric de tres dígits, seguit d´un texte explicatiu.

1xx	L´ordre ha sigut acceptada, però el servidor està pendent de què el client confirmi l´acció
2xx	Operació sol.licitada amb la comanda anterior ha finalitzat amb èxit
3xx	L´ordre ha sigut acceptada, però el servidor està pendent de què el client li envïi noves dades per terminar l´operació
4xx	Resposta d´error, però s´espera que es repeteixi la instrucció
5xx	Error permanent

x0x	Sintaxis
x1x	Informació
x2x	Conexions
x5x	Sistema de correu

Els codis de resposta específics són:

200	(Respuesta no-standard success response, ver rfc876)
211	System status, or system help reply (estado del sistema)
214	Mensaje de Ayuda
220	Servicio Listo
221	Service closing transmission channel
250	Requested mail action okay, completed
251	User not local; will forward to
354	Start mail input; end with .
421	Service not available, closing transmission channel
450	Requested mail action not taken: mailbox unavailable
451	Requested action aborted: local error in processing
452	Requested action not taken: insufficient system storage
500	Syntax error, command unrecognised
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command parameter not implemented
521	does not accept mail (see rfc1846)
530	Access denied because a STARTTLS login is required
550	Requested action not taken: mailbox unavailable
551	User not local; please try
552	Requested mail action aborted: exceeded storage allocation
553	Requested action not taken: mailbox name not allowed
554	Transaction failed

Altres comandes SMTP són:

RSET	Aborta la transacció actual inmediatament, buidant i reiniciant tots els búffers i tables d´estat.
SEND FROM:	[En comptes de MAIL FROM:] El missatge d´envia a la pantalla del terminal de la sessió actual del destinatari del missatge. Si el destinatari no pot rebre el correu, bé perquè no està a la sessió, o perquè el terminal no accepta missatges, etc, el servidor retornarà una resposta.
SOML FROM:	[En comptes de MAIL FROM:] Similar a SEND, amb la diferència que si la pantalla del terminal del destinatari del missatge no pot rebre, pel motiu que sigui, el missatge, la bústia d´aquest usuari rebrà el missatge automàticament.
SAML FROM:	[En comptes de MAIL FROM:] Similar a SOML, amb la diferència de què sempre s´envia el missatge a la bústia independentment de si arriba a la pantalla del terminal o no.
VRFY nomusuari	Verifica que l´usuari especificat existeixi al servidor (abans de començar a enviar-li res, p. ex.)
EXPN [nomllista]	Mostra els noms d´usuari i direccions de correu que pertanyen a la llista de correu especificada
HELP [comanda]	Permet sol.licitar ajuda sobre una comanda
NOOP	Manté viva la conexió
TURN	Sol.licita al servidor que s´intercanvïin els papers: el servidor passarà a ser client i viceversa

NOTA: Els RFCs del protocol SMTP es troben llistats a http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol#Related_Requests_For_Comments
Els RFCs del protocol MIME es troben llistats a http://en.wikipedia.org/wiki/MIME#References
(també és interessant veure: http://www.cicei.com/ocon/gsi/tut_tcpip/3376c47.html)

COURIER:

1.-Instal.lació i Configuració bàsica de Courier-Pop (amb autenticació d´usuaris i utilització de TLS):

El servidor Pop Courier s'instal.la fent un simple *apt install courier-pop* . Aquest servidor ja incorpora autenticació d´usuaris de sèrie (de fet, això forma part de l´estàndar), però aquesta autenticació no està xifrada amb TLS. Per habilitar aquesta possibilitat, haurem d´instal.lar a més els paquets *courier-pop-ssl* i *courier-ssl*

Per tal de què poguem utilitzar el xifratge TLS amb Courier, hem de crear un certificat autosignat. Segurament això es demanarà fer al procés d´instal.lació, però si no és així, simplement haurem d´executar la comanda *mkpop3dcert* i respondre les mateixes preguntes de sempre (o bé utilitzar les respostes que s'hagin escrit prèviament a la secció "[req_dn]" de l'arxiu /etc/courier/pop3d.cnf). El certificat es crearà a /usr/lib/courier/pop3d.pem.

Una altra cosa que s'ha de fer és configurar les bústies dels usuaris existents al sistema. Això es fa executant (com administrador) la ristra de comandes següents dins de la carpeta /etc/courier :

pw2userdb > userdb	(Obtenim la llista d´usuaris del sistema en un format propi de Courier)
chmod 600 userdb	(Aquesta llista només la pot veure el root. Si no, no ens deixarà seguir)
makeuserdb	(Creem la base de dades d´usuaris xifrada a partir de l´arxiu userdb)
maildirmake /home/usuari/Maildir	(Creem la bústia per l´usuari "usuari". S'ha de fer amb tots)
chown -R usuari:usuari /home/usuari/Maildir	(Les bústies creades són propietat de root. Això s´ha de canviar)

En realitat, la comanda maildirmake l'únic que fa és crear la carpeta Maildir amb una sèrie de subcarpetes que és l'estructura necessària per a tenir la bústia ben configurada (aquestes subcarpetes són "tmp", "new" -pel correu no llegit- i "cur" -pel correu ja llegit-). Si ens volem estalviar haver d'executar aquesta comanda per cada usuari nou que es creï al sistema, una possibilitat és fer *maildirmake /etc/skel/Maildir*; d'aquesta manera la bústia es quedaria creada a l'esquelet, de manera que qualsevol nou usuari ja la tindria creada, i sense haver de canviar-ne la propietat.

Finalment, arrancarem els servidors: *systemctl start courier-pop && systemctl start courier-pop-ssl* . A partir d'aquí, podrem accedir a la nostra bústia de forma no xifrada si ens connectem al port 110 (*netcat 127.0.0.0 110*) o de forma xifrada si ens connectem al port 995 (*openssl s_client -connect 127.0.0.1:995* o *ncat –ssl 127.0.0.0 995*).

2.-Instal.lació i Configuració bàsica de Courier-Imap (amb autenticació d´usuaris i utilització de TLS):

Instal.larem els paquets *courier-imap*, i també el *courier-imap-ssl* per utilitzar xifratge.

Igual que passava amb el servidor Pop, per tal de què poguem utilitzar el xifratge TLS amb Courier-Imap, hem de crear un certificat autosignat. Segurament això es demanarà fer al procés d´instal.lació, però si no és així, simplement haurem d´executar la comanda *mkimapdcert* i respondre les mateixes preguntes de sempre. Després només quedarà reiniciar el servidor: *systemctl start courier-imap* i *systemctl start courier-imap-ssl* Després de fer tot això, podrem comprovar com que el nostre servidor Imap està escoltant al port 143 sense xifrar però també al port 993 (xifrat)

Una cosa que podríem fer amb el nostre servidor Imap és configurar carpetes compartides, i permetre que els usuaris creïn ells també les seves pròpies carpetes compartides. Això es faria així:

1.-Com a root, creem un Maildir que es pugui compartir, usant l'indicador -S:
maildirmake -S /var/mail/sysadmins
2.-Creem dins d'aquest Maildir una carpeta compartida on tothom pugui escriure, usant -s:
maildirmake -s write -f unacarpeta /var/mail/sysadmins
Es podria haver creat una carpeta de només lectura fent:
maildirmake -s read -f unaaltra /var/mail/sysadmins
Aquestes carpetes creades estaran ocultes (començaran el seu nom amb un punt)
Aquestes carpetes poden tenir els permisos Linux estàndar com qualsevol altra carpeta
Per esborrar aquestes carpetes, es pot fer amb un simple *rm*
3.-Si un usuari vol compartir una carpeta, pot fer:
maildirmake –add images=/var/mail/sysadmins $HOME/Maildir
Per deixar de compartir (trencar l'enllaç amb el directori compartit):
maildirmake –del images $HOME/Maildir

3.-Instal.lació i Configuració de la interfície web del servidor Courier. Administració de quotes:

Podem administrar tant el servidor Courier-Pop com Courier-Imap amb una còmoda interfície web. Només cal instal.lar el paquet *courier-webadmin*. Compte perquè pot reinstalar algun MPM de l´Apache si el present no és el que el programa necessita. Si apareix una pantalla preguntant si volem activa els scripts CGI, responem afirmativament.

Per utilitzar-lo, cal obrir un navegador i anar a la direcció http://127.0.0.1/cgi-bin/courierwebadmin. Si es vol accedir des d´una altra màquina, només es podrà fer via https, a no ser que es creï un arxiu anomenat "unsecureok" a la carpeta /etc/courier/webadmin

4.-Autenticació d´usuaris en MySQL ó OpenLDAP:

Per defecte, els usuaris del servidor Courier són els usuaris del sistema, ja que per defecte s'utilitza l'autenticació proporcionada pel paquet *courier-authlib-userdb* . No obstant, existeixen altres mètodes d'autenticació proporcionats per altres paquets, com ara Mysql (courier-authlib-mysql) ó un servidor Ldap (courier-authlib-ldap) entre d'altres.

WEBCLIENTS:

Si tenim un servidor Apache+PHP+MySQL ja funcional, podem fer servir el webclient "Webmail-Lite" (https://afterlogic.org/download/webmail-lite-php), les instruccions d'instal.lació del qual es troben a https://afterlogic.com/docs/webmail-lite-8/installation/installation-instructions . També podriem fer servir "RoundCube" (https://roundcube.net) o fins i tot "Rainloop" (http://www.rainloop.net) , el qual no necessita base de dades i les instruccions d'instal.lació del qual es troben a http://www.rainloop.net/docs/installation .

POSTFIX:

Cal tenir configurat prèviament un servidor DNS apuntant a la màquina on s'estigui executant el Postfix. Per tant:

*Si fem servir Bind, hem de tenir l'arxiu /etc/bind/named.conf.local amb el següent contingut …:

```
zone "undominiX.com" IN {
type master;
file "/etc/bind/mevazona.txt";
};
```

...i l'arxiu "mevazona.txt" amb el següent contingut (suposem que la màquina servidora DNS i Postfix és la mateixa i té la IP 192.168.1.1):

```
$ttl 38400
@          IN  SOA servidor.undominiX.com. admin.undominiX.com. (200203194 10800 3600 604800 38400)
@          IN  NS servidor
servidor IN  A   192.168.1.1
smtp      IN  CNAME servidor
@          IN MX 10 servidor
```

*Si fem servir Dnsmasq, hem de tenir l'arxiu /etc/dnsmasq.conf amb el següent contingut:

```
domain=undominiX.com
local=/undominiX.com/
expand-hosts
mx-host=undominiX.com, servidor.undominiX.com,30  #On "servidor.undominiX.com" està a l'arxiu /etc/hosts
```

Per comprovar que la configuració anterior sigui correcta es por realitzar una consulta DNS preguntant pel registre MX, així, per exemple (cal tenir instal.lat el paquet "dnsutils"): *dig @ip.serv.dns undominiX.com MX+short*

Instal.larem el paquet de la forma estàndar: *apt install postfix* . En el procés d'instal.lació ens apareixerà un asistent que ens preguntarà si volem configurar el servidor amb unes opcions predefinides. Li direm que no, que volem configurar-ho "a mà", editant el seu fitxer de configuració, el qual és /etc/postfix/main.cf (propietat de l'usuari root, amb permisos 644). També és molt recomanable instal.lar la documentació oficial: *postfix-doc*

1.-Servidor per la xarxa local (sense autenticació ni xifratge):

Si volem instal.lar un servidor de correu per ús exclusiu de la nostra xarxa interna (sense accés a Internet!), la funció principal del qual seria enviar els missatges a un servidor Pop/Imap també accessible localment (o a un altre servidor Smtp de la mateixa xarxa local), s'hauria d'editar el fitxer de configuració de Postfix amb les següents mínimes línies (on estem suposant que els comptes de correu es corresponen a usuaris reals del servidor SMTP):

#Nom llarg (fqdn) de l'equip. Ha de coincidir amb l'indicat a /etc/hostname i la resolució Dns dels registres MX
***myhostname**=servidor.undominiX.com*
#Nom de domini de l'equip.
***mydomain**=undominiX.com*
#Nom de domini pel correu sortint. Serà el que apareixerà a la dreta de la @ al correu que envïi l'equip
***myorigin** = $mydomain #O el que hi hagi a /etc/mailname*
#Dominis de destí pels que s'acceptarà correu entrant, per repartir-ho localment
***mydestination** = $mydomain localhost.localdomain localhost $myhostname*
#Direccions de remitents autoritzats per utilitzar el servidor per retransmetre correu (fer relay). Per exemple:
***mynetworks** = 127.0.0.0/8 192.168.1.0/24*
#Ruta de la carpeta-bústia dels usuaris, relativa a sa carpeta personal. Un altre valor que pot tenir és "Mailbox" (un #altre sistema més vell on tots els missatges es guarden en un sol fitxer dins de la carpeta /var/spool/mail/usuari)
***home_mailbox** =Maildir/*

> NOTA: La comanda *postconf* mostra els valors de configuració actuals. La comanda *postconf -n* mostra només els valors de configuració que tenen especificat un valor concret diferent del per defecte. La comanda *postconf -m* mostra els mòduls amb els que s'ha compilat Postfix (mysql, ldap, proxy, hash, btree, ...). La comanda *postconf -e "directiva=valor"* serveix per afegir/editar una determinada directiva a l'arxiu /etc/postfix/main.cf sense haver-lo d'editar manualment (per exemple, sudo postconf -e "home_mailbox=Maildir/")

Abans de reiniciar el servei (*systemctl restart postfix*) cal executar la comanda *newaliases*. Aquesta comanda regenera la base de dades interna dels usuaris que Potsfix reconeix com a vàlids (els quals, per defecte, es correspondran als usuaris del sistema).

Aquesta base de dades també pren com entrada el contingut de l'arxiu de text /etc/aliases, el qual es podria editar -opcionalment- per mapejar determinades bústies a unes altres. Per exemple, si es vol que els correus postmaster@undominiX.com i www@undominiX.com (que no estarien associats a cap usuari real del sistema) vagin a parar a la bústia de l'usuari del sistema "root", i que aquesta al seu torn estigui associada a la bústia d'un altre usuari del sistema anomenat "bob", es podrien afegir les línies següents a l'arxiu /etc/aliases:

> *postmaster: root*
> *www:root*
> *root:admin@undominiX.com*

Per provar el servidor, una prova pot ser:

> **1.**-Tenir dos usuaris creats al servidor: un farà de remitent i l'altre de destinatari (encara que l'usuari remitent ens ho podem inventar completament, ja que el nostre servidor encara no ofereix autenticació)
> **2.**-Conectar-s'hi a través de netcat (*nc ip.serv.idor 25*) i enviar algun correu d'un usuari a l'altre
> **3.**-Conectar-se a través de netcat també (*netcat ip.serv.idor 110*) o bé a través de openssl (*openssl s_client -connect ip.serv.idor:995*) al servidor Pop/Pop-ssl que haguem instal.lat a la mateixa màquina per consultar així el correu rebut.

Un cop vist que la prova anterior funciona, la següent prova pot ser:

> **1-**.Tenir dos servidors POP i dos servidors SMTP funcionant en dues màquines diferents ("servidorA" i "servidorB") respectivament, i un servidor DNS autoritatiu de dos dominis funcionant en una màquina qualsevol. La idea és que qualsevol servidor POP rebi missatges originalment enviats des de qualsevol servidors SMTP

2.-Tenir un usuari diferent creat a cada màquina: un serà l'usuari remitent i l'altre (a l'altre màquina) serà el destinatari i viceversa. En realitat, l'usuari remitent pot no existir, ja que el servidor Smtp encara no ofereix autenticació, però l'usuari destinatari sí que ha d'existir per a què la bústia POP sigui funcional.
3.-Conectar-s'hi a través de netcat a "servidorA" (*netcat ip.servi.dorA 25*) i enviar un correu a un usuari destinatari , la bústia del qual està al "servidorB"
4.-Conectar-s'hi a través de netcat a "servidorB" (*netcat ip.serv.idorB 110*) per consultar el correu rebut

2-.Afegir autenticació al servidor SMTP:

Els mètodes d´autenticació SMTP més importants són:

Mètode PLAIN: La comanda per autenticar-se amb aquest mètode és:

AUTH PLAIN Ihlvc295Z2VuaWFsIHBIZG82Nw==

on la ristra de lletres i números final és el resultat d´haver codificat en base64 la cadena "\0nomusuari\0contrasenya" (on"\0" és el caràcter especial null) . Una manera pràctica d'obtenir aquesta ristra corresponent és executar la següent comanda: *echo -ne "\0nomusuari\0contrasenya" | base64* (on el paràmetre -n serveix per no afegir al final de la cadena el caràcter salt de línia i el -e serveix per interpretar correctament el "\0"). Es pot veure que estem utilitzant la comanda *base64*, que és el de/codificador que ve amb els paquet CoreUtils a totes les distribucions Linux (també podem fer servir eines online, com per exemple http://www.elhacker.net/base64.htm ó http://www.motobit.com/util/base64-decoder-encoder.asp , o fer-ho "a mà": l'algoritme no és gaire complicat -veure http://en.wikipedia.org/wiki/Base64 -). Si volguéssim decodificar, hauríem de fer servir el paràmetre -d

Mètode LOGIN: La comanda per autenticar-se amb aquest mètode és: AUTH LOGIN

Seguidament, el servidor contestarà amb una ristra de lletres i números irreconeixible. En realitat, aquesta ristra és la codificació base64 de la cadena "Usuari". El que hem de fer nosaltres és introduir el nostre usuari però codificat en base64. Per fer això, podrem fer servir el mateix mètode anterior, amb la comanda base64 (compte de no afegir al final cap salt de línia ni res!). Un cop fet aquest pas, el servidor ens tornarà a contestar amb una ristra intel.ligible:la codificació base64 de la cadena "Contrasenya". El que hem de fer nosaltres llavors és introduir el nostre password però codificat un altre cop en base64.

Per fer que el nostre servidor Postfix implementi l'autenticació d'usuaris, haurem de fer el següent:

0.-Instal.lar, si no ho estan ja, els paquets "libsasl2-2" , "libsasl2-modules" i "sasl2-bin" . En principi, Postfix està compilat per poder treballar amb aquestes llibreries, però per comprovar-ho es pot fer: *ldd /usr/lib/postfix/sbin/smtpd* i veure si apareixen les llibreries libsasl2, libssl i libcrypto. Si no apareixessin, s'hauria de recompilar el codi font de Postfix . A Fedora els paquets s'anomenen "cyrus-sasl", "cyrus-sasl-plain" i "cyrus-sasl-lib".

1.-Com que el servidor Postfix a l'Ubuntu/Debian està configurat per defecte per funcionar dins d'una gàbia chroot (es pot comprovar que és així si es té a l'arxiu master.cf una línia tal com "smtp inet n - y - - smtpd"), s'han de canviar una sèrie de rutes per a què trobi els fitxers adients dins de la gàbia (que per defecte és /var/spool/postfix, i és on es troben arxius replicats del sistema com ara resolv.conf, hosts, etc). En concret, s'ha de moure el directori on es troben els sockets del dimoni saslauthd dins de la gàbia del Postfix per a què aquest els trobin. Això es fa així:

mkdir -p /var/spool/postfix/var/run/saslauthd
ln -s /var/spool/postfix/var/run/saslauthd /var/run/saslauthd

2.-Afegir l'usuari "postfix" al grup "sasl", per no tenir problemes de permisos: *usermod -a -G sasl postfix*

3.-Modificar l'arxiu /etc/default/saslauthd per a què quedi així:

START=yes
MECHANISMS="pam"
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd"

4.-Modificar l'arxiu /etc/postfix/sasl/smtpd.conf (pot no existir) per a què quedi així

pwcheck_method: saslauthd
mech_list: plain login

(vigilar que no hi hagin espais en blanc al final de les línies ni abans dels dos punts).

5.-Crear l'arxiu /etc/pam.d/smtp amb el següent contingut:

@include common-auth
@include common-account
@include common-password
@include common-session

6.-Afegir les següents línies a l'arxiu /etc/postfix/main.cf:

smtpd_sasl_auth_enable=yes
smtpd_recipient_restrictions=
permit_sasl_authenticated
permit_mynetworks
reject_unauth_destination

7.-Arrencar el servidor SASL: *systemctl start saslauthd* i el servidor Postfix : *systemctl start postfix* Comprovarem que les extensions s'han configurat bé entrant amb netcat al servidor (port 25) i fent un "ehlo".

Cal tenir en compte, no obstant, que en aquest moment només tindríem un sistema d'autenticació sense xifratge, cosa que pot ser un forat de seguretat ja que les contrasenyes viatgen en texte pla (codificades en base64).

Cal tenir en compte també que encara podrem seguir utilitzant el servidor Postfix sense haver d'autenticar-nos per enviar correus destinats al nostre domini (només): això és normal, ja que així es permet rebre correu de l'exterior.

3.-Afegir xifratge al servidor SMTP:

Extensió STARTTLS: A més a més d'autenticar usuaris, una de les extensions que pot implementar un servidor SMTP és la d´utilitzar TLS per xifrar les comunicacions entre client i servidor (normalment, les dues funcionalitats venen juntes, ja que no tindria gaire sentit autenticar un usuari sense xifrar la transmissió de la contrasenya, per exemple). Si en fer un EHLO apareix STARTTLS, estarem en aquest cas.

Per comunicar-nos amb un servidor SMTP que implementi TLS no podrem utilitzar la comanda netcat clàssica ja que aquesta comanda no incorpora la gestió de connexions xifrades. Possibles solucions a això són, per exemple, fer servir un client Netcat que ve dins el paquet "nmap" anomenat *ncat* (que sí permet l'ús de connexions xifrades mitjançant el paràmetre --*ssl*) o bé fer servir com a client la comanda *openssl* directament. La comanda openssl ja sabeu que és una comanda que ens permet fer absolutament de tot amb el tema de la creació i administració de xifratges (simètric i asimètrics), hashes, certificats...però per allò què ens interessa, que és connectar-nos a un servidor SMTP/TLS, simplement haurem de fer: *openssl s_client -connect ipservidorSmtp:port -starttls smtp*

NOTA: Altres opcions serien utilitzar altres versions de Netcat amb TLS incorporat (com per exemple SoCat - http://www.dest-unreach.org/socat -) o bé tunelar NetCat clàssic amb Stunnel - http://www.stunnel.org - , aplicació que precisament permet incorporar xifratge a programes que no el porten de sèrie.

Per fer que el nostre servidor Postfix implementi el xifratge de comunicacions, haurem de fer el següent:

1.-Generar el certificar del servidor (com a root). Es pot fer de moltes maneres (tal com vam veure per exemple a l'hora de muntar un servidor HTTPS), però ara ho farem amb uns assistents que ofereix la pròpia suite Openssl en el cas de l'Ubuntu -en teoria, a Fedora es troba a /etc/pki/tls/misc- :

/usr/lib/ssl/misc/CA.pl -newca
/usr/lib/ssl/misc/CA.pl -newreq
/usr/lib/ssl/misc/CA.pl -sign

El resultat final és la creació dels fitxers "newkey.pem", "newcert.pem" i "demoCA/cacert.pem". S'han de copiar aquests fitxers dins de la carpeta /etc/postfix.

2.-Eliminar la contrasenya de la clau privada que ens han obligat a escriure al pas anterior (amb aquesta contrasenya no es pot iniciar correctament l'Starttls) : *openssl rsa -in newkey.pem -out newkey.pem*

3.-Afegir les línies següents a l'arxiu /etc/postfix/main.cf:

smtpd_use_tls = yes
smtpd_tls_cert_file = /etc/postfix/newcert.pem
smtpd_tls_key_file = /etc/postfix/newkey.pem
smtpd_tls_CAfile = /etc/postfix/cacert.pem
smtpd_tls_received_header = yes

smtpd_tls_session_cache_database = btree:/var/run/smtpd_tls_session_cache
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_loglevel =3
tls_random_source = dev:/dev/urandom

4.-Enviar correus a un altre servidor SMTP d'Internet:

Un MTA, para ser práctico, tiene que ser capaz de reenviar mensajes provenientes de cualquier otro servidor SMTP disponible en Internet y destinados a usuarios pertenecientes a cualquier otro servidor SMTP disponible directamente en Internet. No obstante, un MTA que retransmita correo de cualquiera se dice que está configurado como "open relay", por lo que ser á utilizado masivamente por spammers. Por ello, para controlar la retransmisión de correo proveniente de otros servidores SMTP se utilizan principalmente dos métodos: autenticar los usuarios (tarea que debe realizarse mediante un mecanismo externo, ya que SMTP no provee ningún método de autenticación, siendo SASL es el más utilizado); o bien permitir la retransmisión de determinadas direcciones IP o segmentos de red (es lo que hace la directiva mynetworks).

Por otro lado, cuando se establece una conexión SMTP entre dos MTA, muchos servidores de correo en Internet contrastan la dirección IP del remitente en listas de direcciones IP utilizadas por spammers; si la dirección IP del remitente aparece en esas listas, no se aceptan los mensajes de correo y se cierra la conexión. Si instalamos un servidor de correo en un equipo que accede a Internet con una dirección IP estática y ésta aparece en alguna lista negra, tendremos que seguir una serie de pasos en la configuración para conseguir que saquen nuestra dirección IP de tales listas. Si por el contrario, instalamos un servidor de correo en un equipo con una dirección IP dinámica, esta labor se hace imposible, por lo que hoy día no se puede instalar un servidor de correo que envíe directamente correo en un equipo con dirección IP dinámica.

NOTA: Relacionado con esto está la extensión SPF al protocolo SMTP. Para más información leer https://www.mdirector.com/email-marketing/que-es-el-spf.html

5.-Enviar correus a través de Gmail:

Per tal de superar les dificultats esmentades a l'apartat anterior respecte l'enviament de correu a través d'Internet per part de servidors de correu propis, un possibilitat senzilla podria ser configurar el nostre servidor Postfix per a què no envïi ell directament el correu al servidor destí, sinó que utilitzi l'equip smtp.gmail.com -per exemple, però evidentment pot ser qualsevol SMTP vàlid a Internet- per a què retransmiteixi (relay) els seus missatges. En aquesta configuració, l'equip smtp.gmail.com farà d'"smarthost" del nostre servidor SMTP. Evidentment, haurem de tenir un compte vàlid d'usuari al servidor que farà d'"smarthost".

NOTA: El nom DNS del nostre servidor Postfix podria ser, en aquest cas, oferit perfectament per un servei DynDns o similar, -com ara No-IP (https://www.noip.com)

Els passos són els següents:

0.-Com a mesura de seguretat contra l'spam, per defecte Gmail no permet que l'utilitzin aplicacions per enviar missatges. Per inhabilitar aquesta protecció, inicia sessió manualment dins del compte de Gmail amb què conectarà el Postfix per passar-li els missatges que hagi de reenviar i llavors anar a https://myaccount.google.com/lesssecureapps per activar l'opció "Allow less secure apps"

0BIS.-Assegura't, per si de cas, de tenir els següents paquets instal.lats: "postfix", "mailutils", "libsasl2-2", "ca-certificates" i "libsasl2-modules" (Ubuntu/Debian). A Fedora serien "postfix", "mailx", "cyrus-sasl" i "cyrus-sasl-plan"

1.-Edita l'arxiu /etc/postfix/main.cf per tal de què quedi així:

#587 is the port used by Gmail server.
relayhost = [smtp.gmail.com]:587
#SASL:a standard way to do authentications in email servers (here is to authenticate Postfix in Gmail)
smtp_sasl_auth_enable = yes
#"sasl_passwd" is a file which we'll create to contain password of Gmail account used to send email
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
#TLS:a standard way to send encrypted messages to email servers (Gmail only send encrypted ones)

smtp_use_tls = yes
#To use TLS we need a "CA certificate". Several ones are provided by "ca-certificates" package.
#El fitxer .pem en concret està en https://www.thawte.com/roots/thawte_Premium_Server_CA.pem
#In Fedora, nevertheless, the value of next directive should be /etc/ssl/certs/ca-bundle.crt
smtp_tls_CAfile = /etc/ssl/certs/thawte_Primary_Root_CA.pem

2.-Especificar l'adreça del compte Gmail (i la seva contrasenya) que Postfix usarà per autenticar-se davant de Gmail per tal de poder enviar mails a través d'ell. Això es pot aconseguir executant (com administrador) les següents comandes):

echo "[smtp.gmail.com]:587 username@gmail.com:password" > /etc/postfix/sasl_passwd
postmap /etc/postfix/sasl_passwd
chown root:root /etc/postfix/sasl_passwd* && chmod 600 /etc/postfix/sasl_passwd*

NOTA: La comanda *postmap* genera un fitxer anomenat "/etc/postfix/sasl_passwd.db", el qual "catxeja" el contingut del fitxer "sasl_passwd" en un format binari per tal d'optimitzar la lectura de la informació allà compresa.

3.-Reiniciar Postfix i comprovar que la configuració funciona enviant un correu a algun destinatari que no cal que sigui de Gmail, òbviament. S'hauria de comprovar que el correu hagi arribat

systemctl restart postfix
echo "Text del correu" | mail -s "Assumpte del correu" destinatari@hotmail.com

NOTA: Altres maneres de comprovoar-ho és enviant un mail de prova així: *sendmail -bv username@gmail.com* i mirant els registres: *cat /var/log/mail.log | tail* (també es poden consultar els fitxers /var/log/mail.err, /var/log/mail.warn i /var/log/mail.info). També es pot veure la cua de missatges amb la comanda: *postqueue -p*, (i esborrar tots els missatges de la cua amb *postsuper -d ALL*)

6.-Usar dominis de bústies virtuals en Postfix:

Per tenir comptes d'usuaris en Postfix sense haver de tenir comptes reals Linux es podria utilitzar els dominis de bústies virtuals. Això es fa així:

1.-Afegim a main.cf les següents línies:

virtual_mailbox_domains=undominiX.com undominiY.com unaltre.com #o una ruta de fitxer
virtual_mailbox_base=/var/mail/vhosts
virtual_mailbox_maps=hash:/etc/postfix/vmailbox
virtual_minimum_uid=1000
virtual_uid_maps=static:5000
virtual_gid_maps=static:5000
virtual_alias_maps=hash:/etc/postfix/virtual

2.-Creem o modifiquem l'arxiu /etc/postfix/vmailbox. En aquest arxiu s'emparellen els noms d'usuari amb les seves bústies, que en aquest exemple estan sota el directori /var/mail/vhosts:

unusuari@undominiX.com	undominiX.com/unusuari/
unaltre@undominiX.com	undominiX.com/unaltre/
iunaltre@undominiY.com	undominiY.com/iunaltre/
elmateixambaltrenom@undominiY.com	undominiY.com/iunaltre/

3.-Convertim el fitxer anterior en un arxiu xifrat de taula de búsqueda: *postmap /etc/postfix/vmailbox*

4.-Creem els usuaris, amb Courier, modificant l'arxiu userdb directament, amb el següent format:
unusuari uid=1100 gid=1100|home=/var/mail/vhosts/unusuari|shell=/bin/bash|imappw=|pop3pw=

5.-Generem la contrasenya per cadascun d'aquests nous usuaris amb la comanda *userdbpw -md5* i la copiem/peguem després de les seccions “imappw=” i “pop3pw=” de l'arxiu anterior

6.-Fem *systemctl stop courier-authdaemon* i *makeuserdb*

7.-Configurem Courier per a que utilitzi el fitxer userdb per l'autenticació, a més de les contrasenyes del sistema. Això es fa editant l'arxiu /etc/courier/authdaemonrc i posant la línia:

authmodulelist=”authuserdb” “authpam”

Finalment, reiniciem el servei *courier-authdaemon*