UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
UNAN – LEÓN

Facultad de Ciencias Puras

Departamento de Computación

Tesis para optar al título de
Ingeniero en Sistema de Información

Configuración e Instalación de un completo servidor de correo con Postfix y Cyrus.

Presentado por:

- *Br. Martha María Berríos Reyes*
- *Br. Ana Junieth Blandón González*

Tutor:

Msc. Aldo René Martínez

León, Agosto del 2006

Dedicatoria

A Dios, por haberme dado la fortaleza y la perseverancia de llevar acabo mis estudios.

Mis padres Francisca Reyes y Pedro Berríos que siempre me han apoyado y se han sacrificado porque yo saliera adelante.

Mi tía Cecilia Berríos que me brindo su apoyo y haberme aguantado durante toda la carrera y elaboración de mi tesis.

En especial a mis hermanos que siempre estaban pendientes de mí apoyándome y dándome ánimos para seguir adelante, José Edelberto Berríos Reyes mi súper hermano que siempre estaba pendiente cuando necesitaba algo.

Y en general a mi familia que siempre estaban ahí para apoyar y a mis sobrinos.

Martha María Berríos Reyes

A Dios, nuestro padre celestial por que me demostró todo lo que me quiere y me enseño que él sabe lo que hace, además me dio las fuerzas necesarias para culminar mi carrera y darme la luz en cada uno de los días de trabajo de la elaboración de mi tesis.

A mi mamá Isabel Cristina González Guardado ya que ella me brindo todo su cariño, su apoyo incondicional a lo largo de la carrera.

Mi hermano y mi familia por estar presente siempre dándome ánimos para seguir adelante.

A mi padre Rodolfo Blandón.

Y a unas personas muy especiales como es mi sobrinita Angui Guadalupe González Herrera y Alejandro González.

Ana Junieth Blandón González

Agradecimiento

A Dios que nos dio la fortaleza necesaria para llegar a la culminación de nuestra tesis.

Nuestros padres que siempre nos brindaron su apoyo, su amor en todo momento y comprensión durante todo el tiempo de nuestra preparación hasta llegar a la culminación de nuestra carrera.

A nuestras familias por animarnos a seguir adelante en los momentos más difíciles en los cuales nos sentíamos desanimadas, ellos siempre nos dieron un aliento para continuar y no rendirnos.

Quisiéramos agradecer también a las siguientes personas su colaboración en diversos aspectos que me han llevado a poder realizar nuestra monografía:

Al ingeniero Ranulfo Sánchez Castañeda por darnos un gran apoyo facilitándonos una Laptop para montar nuestro servidor de correo.

Los profesores que nos brindaron sus conocimientos a lo largo de la carrera.

A nuestro tutor Aldo René Martínez, por darnos su tiempo para ayudarnos cuando se nos presentaban problemas en la elaboración de nuestra tesis.

Al administrador del laboratorio numero2 Miltón Torres por ayudarnos a que nos permitieran el acceso al laboratorio para efectuar nuestro trabajo.

A los señores don César y don Ernesto porque siempre nos permitieron el acceso al laboratorio, su apoyo y ánimos para que culmináramos la tesis.

José Ernesto Dávila, ya que nos oriento al principio de nuestra monografía y en algunos momentos cuando se nos presentaban dificultades en el trabajo.

Nuestros compañeros y amigos por todo ese tiempo que compartimos juntos en las aulas de clase y por los momentos de risas.

Y a todas aquellas personas que de una u otra forma estuvieron ahí siempre presente dándonos su apoyo.

Índice

Introducción 1

Objetivos Planteados 3
1.Objetivo General: 3
2.Objetivos Específicos: 3

Antecedentes y Justificación 4

Metodología 5

Recursos Disponibles 6

Marco Teórico 8

1. Correo Electrónico 8
1.1 Elementos de un correo electrónico 9
1.2 Proceso de envío de mensajes 9
1.3 Esquema de transferencia de Correo. 11

2. Elaboración de un servidor de correo electrónico 11
2.1 ¿Qué es Postfix? 12

2.1.1 Razones para utilizar Postfix 12
2.1.2 Características de postfix 13
2.1.3 Característica de seguridad de postfix 13
2.1.4 Arquitectura 13
2.1.5 Tabla 1. Comparación con otros MTA. 15

2.2 Transport Layer Security o TLS 16
2.3 IMAP (Protocolo de Acceso a Mensajes de Internet) 16

2.3.1 Características de IMAP (Internet Message Access Protocol) 17
2.3.2 Características importantes que tiene IMAP y que carece POP3 incluyen: 17

2.4 MIME (Multipurpose Internet Mail Extensions) 19
2.5 Cyrus IMAP (Internet Message Access Protocol). 20
2.6 Cyrus SASL (Simple Authentication and Security Layer) 21
2.7 SMTP (Simple Mail Transfer Protocol) 22
2.8 Protocolo Local de la Transferencia del Correo o LMTP 23
2.9 Amavisd-new 24
2.10 SpamAssassin 24

2.11 Clam Antivirus……25
2.12 Mailman 25
2.13 SquirrelMail 25
2.14 Openssl 27

3. Instalación y configuración del correo. 27
3.1 Esquema general del servidor de correo postfix y cyrus. 28
3.2 Instalación y configuración de Cyrus IMAP y SASL 29
3.3 Instalación del servidor Cyrus IMAP……..30
3.3.1 Configuración de buzones de Correo 35

4. Instalación y configuración de postfix. 37

5. Cifrado del canal de comunicacion usuando TLS 44
5.1 Instalación del paquete openssl 44

6. Correo a través de Web con SquirrelMail. 50
6.1 Instalación de squirrelmail 50

7. Filtros de contenidos. 52
7.1 SpamAssassin 52
7.1.1 Instalación de SpamAssassin. 54
7.2 Clam Antivirus 54
7.2.1 Instalación de Clam Antivirus ..... …….55
7.3 Amavisd-new……..56

8. Medidas anti-UCE……60

9. Configuración de apache2 con SSL 65

10. Listas de correo con Mailman 68

Conclusión..... 71

Recomendaciones..... 72

Glosarios de Términos 73

Los RFC de Internet 83

Bibliografia 84

Anexos 85

Introducción

Durante los últimos años el uso de las tecnologías de información y comunicación en las distintas áreas del sector educativo e investigativo, ha causado un impacto tal que, hoy la gran mayoría de empresas y población en general quieren usar el Internet y las comunicaciones para acelerar sus procesos.

A medida que pasaron los años la información fue ganando mayor importancia en la vida empresarial y en los 60 las grandes compañías comenzaron a instalar grandes computadoras y a conectar terminales a ellas, teniendo así acceso a su información y a sus otros recursos, memoria, procesador, dispositivos de E/S, etc.

Una gran computadora como Mainframe hacía las veces de servidor a las terminales que servía, de ahí que también se le llamara Server (Servidor), dependiendo de los servicios que proporcionara se denominaría File-Server (servidor de archivos) , Print-Server (servidor de impresión). Luego de que los usuarios se familiarizaran con esta nueva metodología de trabajo, se hizo evidente la posibilidad de hacer que los usuarios mismos pudieran dar información a otros usuarios, y hacer así la interacción más dinámica y eficiente, sin la necesidad de que los usuarios tuvieran que estar físicamente juntos, así surgió la implementación de un Mail-Server (servidor de correo).

Postfix creado por Wietse Venema en colaboración con IBM es uno de los mejores servidores de correo del momento. Una de las principales diferencias entre postfix y sendmail es la modularización que permite al programa ser más rápido y eficiente. Además de todo esto Postfix como ya veremos es muy fácil de configurar.

El objetivo de **Postfix** y el servidor **Cyrus IMAP** (Internet Message Access Protocol o Protocolo de Acceso a Mensajes de Internet) es conseguir un sistema de correo electrónico totalmente funcional y de alto rendimiento que use un completo abanico de modernas tecnologías y protocolos que mejoren su eficiencia, robustez, flexibilidad y seguridad. Asimismo, se proporcionan muchas facilidades de uso para los usuarios de este sistema.

MTA: Mail Transport Agent (Agente de Transporte de Correos), el programa encargado de transferir el correo electrónico a través de Internet. No es usado por los usuarios directamente.

MUA: Mail User Agent (Agente de usuario de correo), el programa que ejecuta el usuario para conectarse a un **MTA** y transferirle los mensajes de email. Asimismo se conecta a servidores de mensajería para recepcionar los mensajes enviados al usuario.

Sendmail: Es el **MTA** o Mail Transport Agent (Agente de Transporte de Correos), clásico proporcionado en prácticamente todas las distribuciones Unix. Posee una gran capacidad de configuración mediante reglas y un lenguaje especial de macros.

SMTP: Simple Mail Transfer Protocol o protocolo simple de transferencia de correo electrónico. Es el protocolo estándar de Internet para transferir correo electrónico. Generalmente se utiliza en dos contextos:

1) Para que los **MTA** Mail Transport Agent (Agente de Transporte de Correos), se transfieran mensajes mutuamente.

2) Para que los **MUA** Mail User Agent (Agente de usuario de correo) remitan mensajes al **MTA** Mail Transport Agent (Agente de Transporte de Correos).

Objetivos Planteados

Objetivo General:

- Instalar y configurar un servidor de correo con **Postfix** y **Cyrus**, con sus políticas de filtro y seguridad asociadas.

Objetivos Específicos:

- Instalar y configurar un servidor de correo bajo el subdominio de la UNAN-LEON.
- Configurar un servidor el cual funcione para la universidad como para cualquier otra institución.
- Implementar y probar la funcionalidad del servidor, con los mecanismos de seguridad asociados.
- Documentar el proceso de instalación y configuración del servidor de correo.

Antecedentes y Justificación

Hoy en día no contamos con documentos relacionados a la configuración e instalación de servidor de correo electrónico, por lo cual nosotras hemos decido realizar nuestro proyecto en base ha este tema, además es una forma de brindar un aporte al gremio del Departamento de Computación.

Se pretende obtener un sistema de correo con las siguientes características:

- Independencia de los usuarios de sistema y las cuentas de correo electrónico.
- Un dominio principal donde se crean cuentas de correo.
- Autenticación a través de **SASL** (Simple Authentication and Security Layer o Capa de Seguridad y Autenticación Simple), con métodos de texto plano o login.
- Transporte seguro del tráfico mediante **TLS** (Transport Layer Security o Seguridad para Capa de Transporte).
- Acceso a los buzones por **IMAP** (Internet Message Access Protocol o Protocolo de Acceso a Mensajes de Internet) sobre **SSL** Simple Authentication and Security Layer (Capa de Seguridad y Autenticación simple) y por webmail.
- Filtros antivirus con **ClamAV** y antispam **SpamAssassin**.

También de esta forma obtendremos nuevos conocimientos que nos servirán en un futuro, el cual no adquirimos durante el transcurso de la carrera.

Metodología

El método que utilizamos durante el desarrollo de nuestro proyecto es el Método Científico General, que ayuda a establecer la estrategia de la investigación, ya que es una metódica ordenada y coherente.
Los mecanismos ha utilizar serán:

- Consultas bibliografícas.
- Pruebas experimentales.
- Entrevistas con expertos en la materia.
- Método del ciclo de vida.

Es por ello que nuestro trabajo asume la metodología expuesta anteriormente.

Recursos Disponibles

Los recursos que utilizamos para la elaboración del proyecto son:

Dos computadoras con las siguientes características:

- ✓ Procesador: Pentium IV
- ✓ Memoria Ram: 256Mb
- ✓ Disco duro: 20GB.
- ✓ Monitor: hp 54

Sistemas operativos instalados:

- ✓ **Ubuntu Linux (2.6.12-9-386):** Ubuntu es un sistema operativo, basado en el núcleo Linux. La estructura técnica del sistema está basada en el Proyecto Debian, pero el ideario está inspirado en los principios de la corriente Ubuntu, palabra africana que significa "humanidad hacia los demás"

Sistema operativo Linux.

El sistema operativo **Linux** es un sistema operativo del tipo Unix para ordenadores PC y compatibles.

Es un sistema multiusuario con multitareas real y está en constante evolución, todas las semanas se incorpora algo nuevo al sistema Operativo.

El sistema operativo **Linux** es sistema que se distribuye bajo licencia **GPL** (General Public License o Licencia Pública General), lo que significa que puede ser libremente distribuido, copiado y modificado.

Algunas de las principales características y por las cuales se planteó el uso de este S.O Linux son:

- Multitarea, varios programas funcionando a la vez.
- Multiusuario, varios usuarios en la misma máquina a la vez.
- Tiene protección de memoria entre procesos, por lo que un programa no puede colgar el resto del sistema.
- Paginación compartida de memoria “copy-on-write”, copia al escribir. Este es un eficiente método donde se comparte una página o un recurso hasta que se logra el intento de escritura. En ese caso se hace

una copia, y la escritura se realiza en la copia entre ejecutables.

- Uso de librerías dinámicas compartidas (DLL).
- Swapping de memoria virtual en disco.
- Ampliamente compartible con POSIX, System V y BSD a nivel de código fuente y con 1SCO a través de un módulo de emulación.
- Todo el código fuente del sistema y sus aplicaciones están disponibles, incluido el del Kernel y drivers.
- Control de tareas POSIX (Potable Operating System Interface for UNIX).
- Emulación del 387 en el kernel, por lo que los programas no necesitan incluir una emulación propia. Todos los ordenadores en los que se ejecuta Linux se compartan como si tuviesen coprocesador matemático en caso que no lo tengan. Por supuesto, si tu CPU incluye coprocesador, este será usado en lugar de la emulación, e incluso puede recompilar tu propio kernel con la parte de la emulación eliminada para ganar esa parte de la memoria.
- Soporta los teclados de la mayoría de los países, con sus acentos, etc, y es fácil añadir nuevos.
- Múltiples terminales virtuales en uno, a través de combinaciones de teclas disponen de hasta 64 terminales, cada uno con una sesión distinta.
- Soporta múltiples sistemas de ficheros, incluyendo minix-1, Xenix y todos los sistemas de ficheros de System V. Por supuesto incluye un sistema de ficheros propios permitiendo capacidades de hasta 4 Tb (Terabytes) y nombres de hasta 255 caracteres.
- Un modo especial llamado UMSDOS permite a Linux instalarse en una partición DOS normal.
- Soporte de particiones PSF-DOS de OS/2 2.1 en modo lectura.
- Completo soporte de todos los estándares de CD-ROM.
- En red: TCP/IP incluyendo ftp, telnet, NFS, etc.

✓ **Windows XP con Service Pack 2.0**

Marco Teórico

1. Correo Electrónico

El e-mail uno de los servicios de Internet más populares en la actualidad.

Con el crecimiento de Internet en los últimos años, se han producido cambios fundamentales en el correo electrónico. Ya no es sólo una herramienta interna en las empresas y organizaciones. Actualmente, sirve de unión entre las personas de distintas compañías y países, y ha permitido que desde la tierra se comparta información con las personas que están en el espacio, como si se encontraran en el mismo edificio.

Podría decirse que el correo electrónico se ha convertido en el beneficio más importante de Internet hasta la fecha. Dada cada vez mayor integración del correo electrónico en la vida de las personas y las empresas, su importancia aumenta diariamente. Si antes el correo electrónico era una comodidad, ahora es una necesidad. En el pasado, las personas utilizaban el correo electrónico simplemente para enviarse notas cortas sin importancia unos a otros. Sin embargo, ahora se utiliza para enviar información crucial.

El crecimiento sin precedentes del correo electrónico ha sido posible gracias a la adopción mundial del protocolo o lenguaje subyacente del correo electrónico de Internet, el Protocolo simple de transferencia de correo**.** El estándar **SMTP** (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico) hace posible que diferentes sistemas de correo electrónico conectados a Internet intercambien información entre sí.

Sin embargo, a pesar de todas las ventajas que **SMTP** (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico) ha aportado a Internet, adolece de un problema inherente. El estándar **SMTP** (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico) fue desarrollado originalmente para transportar mensajes breves de relativamente poca importancia en una red cerrada, no para transportar información crucial y confidencial en un mundo interconectado. Ninguna de las personas que desarrollaron el protocolo **SMTP** (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico) podía imaginar que desempeñaría el papel que tiene en la actualidad. A causa de ello, **SMTP** (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico) no fue diseñado para proteger el tipo de información que transporta hoy a través de las redes por las que pasa. Fue diseñado para transportar información más sencilla a través de redes más simples, de ahí el nombre Protocolo simple de transferencia de correo. Por ejemplo, **SMTP** envía información a través de Internet de forma que todos pueden leer el mensaje.

1.1 Elementos de un correo electrónico

Para que una persona pueda enviar un correo a otra, ambas han de tener una **dirección de correo electrónico**. Esta dirección la tiene que dar un **proveedor de correo**, que son quienes ofrecen el servicio de envío y recepción. El procedimiento se puede hacer desde un **programa de correo** o desde un **correo Web**.

- **Dirección de correo**

Una **dirección de correo electrónico** es un conjunto de palabras que identifican a una persona que puede enviar y recibir correo. Cada dirección es única y pertenece siempre a la misma persona.

Un ejemplo es **martha@marana.isi.unanleon.edu.ni**, que se lee martha arroba marana punto isi punto unanleon punto edu punto ni. El signo @ (llamado **arroba**) siempre está en cada dirección de correo, y la divide en dos partes: el nombre de usuario (a la izquierda de la arroba; en este caso, **martha**), y el **dominio** en el que está (lo de la derecha de la arroba; en este caso, **marana.isi.unanleon.edu.ni**). La arroba también se puede leer "en", ya que **martha@marana.isi.unanleon.edu.ni** identifica al usuario martha que está **en** el **servidor marana.isi.unanleon.edu.ni** (indica una relación de pertenencia).

Una dirección de correo se reconoce fácilmente porque siempre tiene la @. Lo que hay a la derecha de la arroba es precisamente el nombre del *proveedor* que da el correo, y por tanto es algo que el usuario no puede cambiar. Por otro lado, lo que hay la izquierda normalmente sí que lo elige el usuario, y es un identificador cualquiera, que puede tener letras, números, y algunos signos.

- **Proveedor de correo**

Para poder usar enviar y recibir correo electrónico, generalmente hay que estar registrado en alguna empresa que ofrezca este servicio (gratuita o de pago). El registro permite tener una *dirección de correo* personal única y duradera, a la que se puede acceder mediante un nombre de usuario y una contraseña.

1.2 Proceso de envío de mensajes

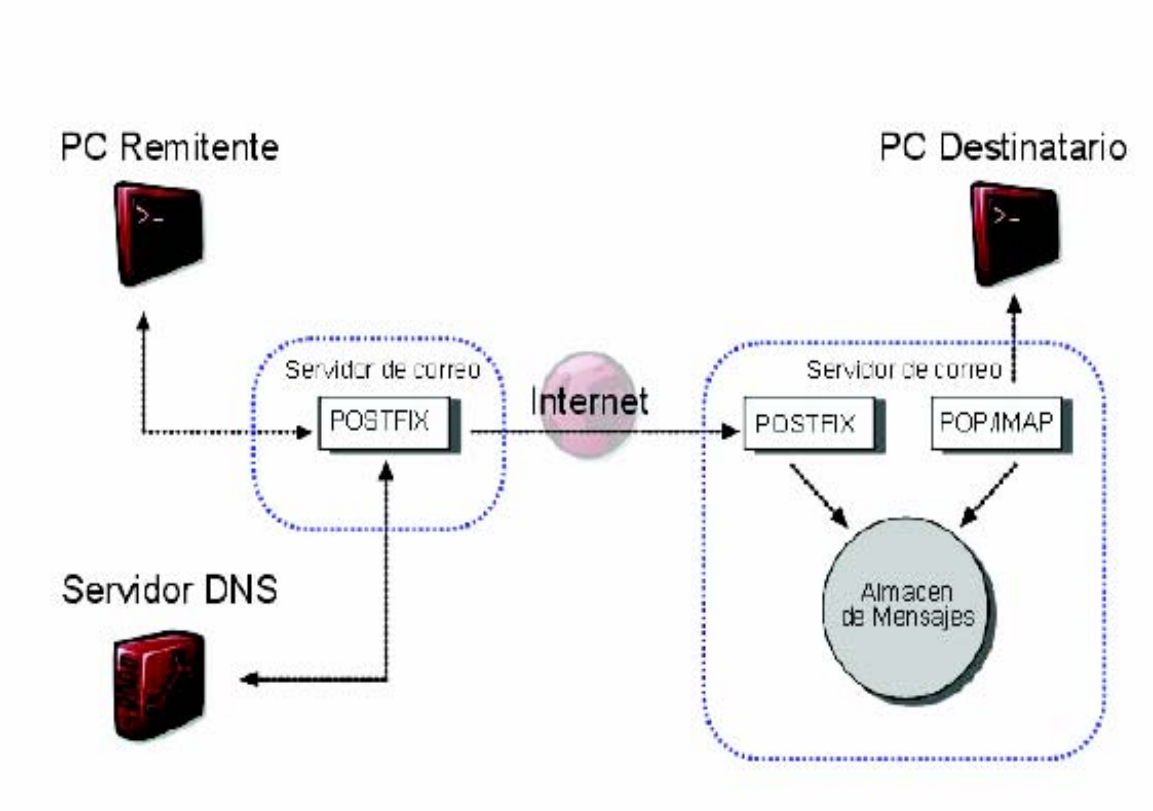
El e-mail comenzó como la posibilidad que permitía a distantes colegas que trabajaban para una empresa que tenía una LAN trabajar juntos, compartir experiencia, e intercambiar ideas y proyectos, luego se vislumbró la posibilidad de hacer que un usuario pudiera acceder a este mismo servicio en

forma remota es decir sin estar conectado a la red, en realidad conectado por medio de una línea telefónica y un MODEM.

El proceso de envío de un mensaje de correo, consistía originalmente en un usuario escribiendo el mensaje en un programa de aplicación llamado cliente de correo, en contraposición con el servidor de correo, que consistía de un editor de texto, posiblemente un corrector ortográfico, una base de datos de la forma de una libreta de direcciones, un administrador de archivos (los mensajes recibidos o no enviados) y un módulo de comunicaciones para poder transferirlos.

El mensaje quedaba almacenado en el mail-server hasta que el usuario destinatario usando su cliente de correo se conectara con él y solicitara los mensaje que le tuviera reservados, el proceso inverso de envío de mensajes era muy parecido cuando el usuario terminara de escribir su mensaje, especificando la dirección de el destinatario, se conectaba con el servidor a fin de depositar el archivo hasta que el destinatario lo solicitara. Cuando el servidor está conectado a sólo una red la única limitación de la dirección de destino, además de no permitir espacios en blanco en la dirección, era que cada dirección debía identificar de forma unívoca a cada usuario, con una LAN esta restricción es fácil de implementar pero con más de una ya pasa a ser un problema mayor; así se introducen los dominios de los usuarios que representan a que servidor pertenecen y que tienen la forma de una dirección válida, es decir sin espacios en blanco ni caracteres prohibidos, para diferenciar el nombre del usuario de su dominio se adoptó en carácter "@" que significa "en" (at) entonces la dirección Bruno@Servidor.A se puede leer como "Bruno en Servidor.A"

1.3 Esquema de transferencia de Correo.
Esquema 1



2. Elaboración de un servidor de correo electrónico

Hoy en día para la comunicación a grandes distancia se hace necesaria la creación de un servidor de correo electrónico el cual será útil para la comunicación de las persona.

El **MTA** (Mail Transportation Agent o Agente de Transporte de Correos) **Postfix** pretende ser rápido, fácil de administrar y seguro, a la vez que suficientemente compatible con Sendmail como para que los usuarios existentes no se asusten. Por lo tanto, externamente mantiene el estilo de Sendmail, mientras que internamente es completamente diferente.

A diferencia de **Sendmail**, **Postfix** no es un programa monolítico, sino una combinación de pequeños programas, cada uno de los cuales lleva a cabo una función especializada.

2.1 ¿Qué es Postfix?

Postfix es un programa al que podemos llamar daemon (demonio) que esta en escucha en nuestra máquina en un determinado puerto (25) y que cumple determinadas funciones respondiendo a ciertos datos de entrada que recibe.

Postfix es un servidor de correo, un daemon, que gestiona la entrada y la salida de correos de Internet a la intranet o de la intranet a Internet o sin salir de la propia intranet. **Postfix** fue diseñado por Wietse Venema como alternativa a sendmail. **Postfix** como rige el estándar del protocolo **smtp** (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico) rfc2821 se instalará en el puerto 25 por defecto y ahí empezará a funcionar con las opciones por defecto salvo que le digamos lo contrario.

2.1.1 Razones para utilizar Postfix

Las razones para usar **Postfix** fueron básicamente su sencillez, potencia y versatilidad a respuesta a todas las interrogantes, porque es tan potente como sendmail, fácil de configurar y además, hasta es entretenido.

- Diseño modular (no es un único programa monolítico).
- La seguridad ha sido un condicionante desde el comienzo de su diseño.
- Lo mismo cabe decir del rendimiento (seguramente Sendmail no se diseñó pensando que algún día habría sitios necesitaran procesar cientos de miles o millones de mensajes al día).
- Soporte para las tecnologías más usadas hoy día: LDAP, Bases de datos (MySQL), autentificación mediante **SASL** Simple Authentication and Security Layer (capa de seguridad y autenticación simple), **LMTP** (Local Mail Transfer Protocol o Protocolo de la Transferencia del Correo Local), es un derivado del **smtp**, el Simple Mail Transfer Protocol., etc.
- Estricto cumplimiento de los estándares de correo-e.
- Facilidad de configuración.
- Abundante documentación y de calidad.
- Fácil integración con antivirus.
- Uso sencillo de listas negras.
- Tiene múltiples formas de obtener información de lo que está pasando para resolver problemas o simplemente, para aprender.
- Se pueden lanzar varias instancias de **Postfix** en la misma máquina con distintas configuraciones, usando cada una distinta direcciones IP, distintos puertos, etc.
- Filtrado de cabeceras y cuerpos de mensajes por expresiones regulares.
- Utilidades para varias cosas, como gestionar las colas de mensajes.

Por último, pero no menos importante, hay que decir que el código fuente de **Postfix** (por supuesto de dominio público) es un ejemplo de diseño, claridad y documentación, lo cual facilita su mantenimiento, así como la incorporación de nuevas capacidades, corrección de errores, etc.

2.1.2 Características de postfix

- Servidor de correo que funciona sobre sistemas de tipo Unix.
- Su intención fue la de sustituir a sendmail. Compatible para el resto de aplicaciones.
- Arquitectura y diseño muy modular.
- Fácil de administrar y configurar.
- Repartir correo de forma local puede repartir a almacén de correo o pasarlo a un **MDA** (Mail Delivery Agent o Agente de Entrega de Correo).
- Muy rápido. Fue diseñado pensando en el rendimiento. Evita saturar otros sistemas.

2.1.3 Característica de seguridad de postfix

- Arquitectura modular: Cada proceso se ejecuta con privilegios mínimos para su tarea.
- Proceso que no se necesita se deshabilita: No se puede explotar.
- Los procesos se aíslan unos de otros. Muy poca comunicación entre procesos.
- Evita utilizar buffers de tamaño fijo, evitando que tengan éxito ataques buffer overflow.
- Puede ejecutarse en modo chroot.
- Preparado para ataques DoS (Deny of Service, Denegación de Servicio). Cantidad de memoria controlada.

2.1.4 Arquitectura

1. Colas de correo

Postfix basa su funcionamiento en cuatro colas: maildrop, incoming, active y deferred.

- **Maildrop queue:** El correo que es generado y/o entregado localmente en el sistema es procesado por la cola Maildrop.

- **Incoming queue o cola entrante:** Esta cola recibe correo de otros hots, clientes o de la cola maildrop. Si llegan correos y postfix no puede atenderlos se quedan esperando en esta cola.

- **Active queue o cola activa:** En esta cola están los mensajes en la fase de encaminamiento.

- **Deferred queue o cola diferida:** En esta cola se almacena los mensajes que no se han podido encaminar o están pendientes de reintentar su encaminamiento.

2. Procesos

Postfix gestiona las colas mediante procesos independientes.

- **Pickup o recolección:** Recoge los correo que provienen de las cola maildrop y los pasa a cleanup.

- **Smptd:** Este proceso atiende, mediante el protocolo SMTP los correos de otros sistemas.

- **Cleanup o limpieza:** Analiza las cabeceras de los correos. Si es ok. Los deposita en la cola incoming.

- **Qmgr:** Proceso encargado de tratar los correos que llegan a incoming, depositarlos en active y lanzar el proceso adecuado para su encaminamiento: local, smtp o pipe.

- **Local:** Proceso encargado de depositar el correo en el buzón.

- **Smtp (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico):** Proceso encargado de enviar el correo al host destino mediante protocolo **SMTP**.

3. Comandos

Algunos comandos de **Postfix** más interesantes:

- **newaliases:** Actualiza la base de datos de las alias (/etc/aliases). Enlace simbólico a sendmail (compatibilidad).

- **postsuper:** Se encarga de realizar operaciones de mantenimiento.

- **postqueue:** Comando que sirve de interfaz para la gestión de las colas.

- **postmap:** Crea, actualiza o consulta una o más tablas postfix.

- **postconf:** Muestra los valores actuales de los parámetros de postfix.

4. Tablas

Las tablas, creadas por el administrador sirven a los procesos para saber que tratamiento hay que dar a cada correo. Son 6 tablas aunque no son obligatorias.

- ✓ **Access:** Sistemas a los que se acepta o rechaza los correos. La utiliza el proceso smtpd.
- ✓ **Aliases:** Define nombres alternativos a usuarios locales. Consulta el proceso local.
- ✓ **Canonical:** Relación entre nombres alternativos y reales, locales o no. Proceso cleanup.
- ✓ **Relocated:** Devolver los mensajes que han cambiado de dirección. Proceso qmgr.
- ✓ **Transport**: Política de encaminamiento por dominios. Proceso trivial-rewrite.
- ✓ **Virtual:** Relación entre usuarios virtuales y reales. Proceso cleanup.

Postfix soporta muy diversos soportes de backend para las tablas.

2.1.5 Tabla 1. Comparación con otros MTA (Agentes de Transporte de correo).

MTA	Desarrollo	Seguridad	Características	Rendimiento	CompSendemail	Modular
Qmail	Normal	alta	Altas	Alto	complementos	si
Sendemail	Alto	Baja	Altas	Bajo	x	no
Postfix	Bajo	Alta	Normales	Alto	si	si
Exim	Normal	Baja	Altas	Normal	si	no

CompSendmail: Significa que el **MTA** Mail Transport Agent (Agente de Transporte de Correos), se comporta como Sendmail en algunos aspectos que harán que sea más transparente cambiarse de Sendmail a un agente alternativo de transporte de correo.

2.2 Transport Layer Security o TLS (Capa de Transporte Segura)

Por defecto, toda comunicación en Internet se hace sin ningún tipo de cifrado y sin una autenticación fiable. Esto significa que cualquiera con acceso físico a la línea de datos por la que viaja un paquete puede espiar dicha comunicación. Aún peor, es posible redirigir o alterar esa comunicación para que la información que se desea mandar se pierda y nadie se dé cuenta.

De cara a solventar estos problemas de seguridad, Netscape, Inc. introdujo el protocolo **SSL** (Secure Sockets Layer), que ha ido evolucionando en el protocolo estandarizado **TLS** (Transportation Layer Security). Ofrece tanto cifrado de la comunicación (frenando las escuchas) como autenticación fuerte (asegurando que ambas partes de una comunicación son correctamente identificadas y que la comunicación no puede ser alterada).

Postfix/TLS no implementa el protocolo **TLS** por sí mismo, sino que usa el paquete **OpenSSL** para esta tarea. Mejora la comunicación TCP añadiendo cifrado e integrada en los correos, no protege el contenido de los mails. Necesita un par de claves públicas y privadas y Autoridad Certificadora (CA).

2.3 Internet Message Access Protocol o IMAP (Protocolo de Acceso a Mensajes de Intenet).

IMAP es un acrónimo inglés de Internet Message Access Protocol, lo cual indica que es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

Mediante **IMAP** se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet. Una vez configurada la cuenta IMAP, puede especificar las carpetas que desea mostrar y las que desean ocultar, esta característica lo hace diferente del protocolo **POP** (Post Office Protocol o Protocolo de Oficina de Correos).

Protocolo diseñado con el fin de permitir la manipulación de buzones remotos como si fueran locales. **IMAP** requiere de un servidor que haga las funciones de oficina de correos pero en lugar de leer todo el buzón y borrarlo, solicita sólo los encabezados de cada mensaje. Se pueden marcar mensajes como borrados sin suprimirlos completamente, pues estos permanecen en el buzón hasta que el usuario confirma su eliminación. Ofrece soporte para diferentes modos de acceso: online, offline, desconectado.

2.3.1 Características de IMAP (Internet Message Access Protocol o Protocolo de Acceso a Mensajes de Intenet)

- Es un servidor que da acceso **IMAP** a los Maildirs. Esta diseñado para ello.
- Servidor **IMAP** incluido en Courier Mail Server. Configurado en solitario puede trabajar con otros **MTA's** que reparten Maildirs.
- Soporta varias extensiones al formato Maldir básico como carpetas y quotas por soft.
- Incluye módulos de autentificación abstractos. Passwd, PAM, MySQL, PostreSQL, LDAP…
- Ofrece **IMAP** sobre **SSL** (Secure Sockets Layer). Soporte IPv6.
- Soporta carpetas compartidas entre grupos de usuarios.
- Permite limitar el número de accesos de **IMAP** y numero máximo de accesos desde la misma IP.
- Escrito en C.
- A diferencia del protocolo **POP3** (Post Office Protocol o Protocolo de Oficina de Correos), el correo no es descargado inmediatamente sino que es leído o consultado directamente en el servidor.
- Permite ver únicamente los encabezados del mensaje antes de decidir si abrirlo o eliminarlo.
- El servidor retiene el correo hasta que se solicite su eliminación.
- Puede consultarse el mismo correo desde diferentes computadoras ya que solo se lee lo que hay en el servidor.
- Permite operaciones avanzadas como creación de carpetas y buzones en el servidor.

IMAP fue diseñado como una moderna alternativa a **POP** fundamentalmente, los dos protocolos les permiten a los clientes de correo acceder a los mensajes almacenados en un servidor de correo.

2.3.2 Características importantes que tiene IMAP y que carece POP3 incluyen:

Soporte para los modos de operación *connected* y *disconnected*:

Al utilizar **POP3**, los clientes se conectan al servidor de correo brevemente, solamente lo que les tome descargar los nuevos mensajes. Al utilizar **IMAP**, los clientes permanecen conectados el tiempo que su interfaz permanezca activa y descargan los mensajes bajo demanda. El patrón de **IMAP** puede dar tiempos de respuesta más rápidos para usuarios que tienen una gran cantidad de mensajes.

- Soporte para la conexión de múltiples clientes simultáneos a un mismo destinatario.

  El protocolo **POP3** asume que el cliente conectado es el único dueño de una caja de correo. En contraste, el protocolo **IMAP** permite accesos simultáneos a múltiples clientes y proporciona ciertos mecanismos a los clientes para que se detecten los cambios hechos a un mailbox por otro cliente concurrentemente conectado.

- Soporte para acceso a partes **MIME** (Multipurpose Internet Mail Extensions, Extensiones de Correo Internet Multipropósito) de los mensajes y obtención parcial.

  Casi todo el email del Internet es transmitido en formato **MIME** (Multipurpose Internet Mail Extensions, Extensiones de Correo Internet Multipropósito). El protocolo **IMAP** les permite a los clientes obtener separadamente cualquier parte **MIME** individual así como, obtener porciones de las partes individuales ó los mensajes completos.

- Soporte para que la información de estado del mensaje se mantenga en el servidor.

  A través de la utilización de banderas definidas en el protocolo **IMAP** de los clientes, se puede vigilar el estado del mensaje, por ejemplo, si el mensaje ha sido o no leído, respondido o eliminado. Estas banderas se almacenan en el servidor, de manera que varios clientes conectados al mismo correo en diferente tiempo pueden detectar los cambios hechos por otros clientes.

- Soporte para accesar múltiples buzones de correo en el servidor.

  Los clientes de **IMAP** pueden crear, renombrar o eliminar correo (por lo general presentado como carpetas al usuario) del servidor, y mover mensajes entre cuentas de correo. El soporte para múltiples buzones de correo también le permite al servidor proporcionar acceso a los fólders públicos y compartidos.

- Soporte para búsquedas de parte del servidor.

  IMAP proporciona un mecanismo para los clientes le pidan al servidor que busque mensajes de acuerdo a una cierta variedad de criterios. Este mecanismo evita que los clientes descarguen todos los mensajes de su buzón de correo con el fin de agilizar las búsquedas.

- Soporte para un mecanismo de extensión definido.

  Como reflejo de la experiencia en versiones anteriores de los protocolos de Internet, **IMAP** define un mecanismo explícito mediante el cual puede ser extendido. Se han propuesto muchas extensiones de **IMAP** y son de uso común. Un ejemplo de extensión es el **IMAP IDLE**, que sirve para que el servidor avise al cliente cuando ha llegado un nuevo mensaje de correo y éstos se sincronicen. Sin esta extensión, para realizar la misma tarea el cliente debería contactar periódicamente al servidor para ver si hay mensajes nuevos.

IMAP es utilizado frecuentemente en redes grandes; por ejemplo los sistemas de correo de un campus. **IMAP** les permite a los usuarios acceder a los nuevos mensajes instantáneamente en sus computadoras, ya que el correo está almacenado en la red. Con **POP3** los usuarios tendrían que descargar el email a sus computadoras o accesarlo vía Web. Ambos métodos toman más tiempo de lo que le tomaría a **IMAP**, y se tiene que descargar el email nuevo o refrescar la página para ver los nuevos mensajes.

De manera contraria a otros protocolos de Internet, **IMAP** soporta mecanismos nativos de cifrado. La transmisión de contraseñas en texto plano también es soportada.

2.4 MIME (Multipurpose Internet Mail Extensions, Extensiones de Correo Internet Multipropósito)

Se trata de un Standard que especifica como debe un programa (inicialmente un programa de correo o un navegador Web) transferir archivos multimedia (video, sonido, por extensión cualquier archivo que no esté codificado en US-ASCII). **MIME** adjunta un archivo de cabecera a cada archivo, especificando el tipo y el subtipo del contenido del archivo principal. Gracias a esta información tanto el servidor como el navegador pueden manejar y presentar correctamente los datos además que trabaja en el nivel de presentación del modelo OSI.

En la actualidad ningún programa de correo electrónico o navegador de Internet puede considerarse completo si no acepta **MIME** en sus diferentes facetas (texto y formatos de archivo).

Su funcionamiento se basa en:

1.- Clasificar los contenidos a transmitir según diversos tipos.

2.- Establecer que acción se toma para cada tipo de fichero que se transmite.

Es decir, que tipo de codificación debe utilizarse para cada fichero.

Una de las características principales de los sistemas de codificación **MIME** es que se han diseñado de forma que se mantenga sin modificación la mayor parte posible de texto (todos los caracteres que sean US-ASCII se transmitan sin modificación), solo son codificados aquellos caracteres "no estándar" que puedan tener problemas en alguna parte del sistema Internet. De esta forma, si algún agente de correo no es conforme a **MIME**, el texto codificado todavía tendrá algún sentido (lo que ocurre cuando recibimos, o nos dicen que reciben, esos caracteres extraños intercalados en nuestros correos).

2.5 Cyrus IMAP (Internet Message Access Protocol o Protocolo de Acceso a Mensajes de Internet).

Cyrus IMAP es desarrollado y mantenido por el Andrew Systems Group de la Carnegie Mellon University.

Es un potente servidor de correo y news que soporta **IMAP**, **IMAPS**, y el lenguaje de scripts de filtrado de mensajes y respuestas sieve para crear servidores de alta disponibilidad. Las ventajas que tiene un servidor imap frente a uno pop son considerables. Por una parte, las carpetas se crean en el servidor, no en el cliente, por lo que es fácil realizar un backup de todas las carpetas de los usuarios sin tener que tocar los clientes. Por otra parte, al estar en el servidor es sencillo configurar diferentes modos de acceso a las carpetas que permiten que con una sola autentificación imap podamos ver todas las cuentas asociadas en caso de que un usuario tenga más de una. Además podemos establecer cuotas, para limitar el espacio en disco que ocupará el correo del usuario.

A diferencia de otros servidores **IMAP**, **Cyrus** usa su propio método para almacenar el correo de los usuarios. Cada mensaje es almacenado en su propio fichero. El beneficio de usar ficheros separados es una mayor fiabilidad ya que sólo un mensaje se pierde en caso de error del sistema de ficheros. Los metadatos, tales como el estado de un mensaje (leído, etc.) se almacenan en una base de datos. Además, los mensajes son indexados para mejorar el rendimiento de **Cyrus**, especialmente con muchos usuarios e ingentes cantidades de mensajes.

Otra característica muy importante es que no son necesarias cuentas locales de Linux para cada usuario. Todos los usuarios son autenticados por el servidor **IMAP**. Esto lo convierte en una magnífica solución cuando se tiene una gran cantidad de usuarios.

La administración es llevada a cabo mediante comandos especiales de **IMAP**. Esto le permite usar tanto la interfaz de línea de comandos como las interfaces Web. Este método es mucho más seguro que un interfaz Web para /etc/passwd.

Desde la versión 2.1 de **Cyrus**, se usa la versión 2 de la librería **SASL** para la autenticación. Cyrus se autentica con saslauthdaemon, quien redirige la petición al mecanismo que le hayamos definido.

El servidor **IMAP Cyrus** está generalmente orientado para ser ejecutado en sistemas sellados, en los cuales los usuarios normales no tienen acceso permitido para identificarse.

La base de datos de la cuenta de correo está almacenada en partes donde el sistema de archivos es privado y seguro; controlado por **Cyrus**.

El servidor **IMAP** hará uso de los protocolos **IMAP, POP3** o **KPOP**.

El diseño privado de la base de datos de correo provee a ésta de una gran eficiencia, escalabilidad y grandes posibilidades de administración.

La aplicación permite múltiples conexiones de lectura y escritura simultáneas, soporte para listas de control, múltiples mecanismos **SASL** y reglas de filtrado para correo.

2.6 Cyrus SASL (Simple Authentication and Security Layer o Capa de Seguridad y Autenticación Simple)

SASL es un método para añadir soporte para la autenticación a protocolos basados en la conexión que ha sido estandarizado por la IETF (Internet Engineering Task Force). Se usa en servidores (en este caso **Cyrus IMAP**) para manejar las peticiones de autenticación de los clientes. Para ello, el protocolo incluye un comando para identificar y autenticar un usuario contra un servidor y para, opcionalmente, negociar la protección de las subsiguientes interacciones del protocolo. Si se negocia su uso, una capa de seguridad es añadida entre el protocolo y la conexión.

Cyrus SASL es una implementación de **SASL** que puede ser utilizada del lado del servidor o del lado del cliente y que incluye como principales mecanismos de autenticación soportados a ANONYMOUS, CRAM-MD5, DIGEST-MD5, GSSAPI y PLAIN. El código fuente incluye también soporte para los mecanismos LOGIN, SRP, NTLM, OPT y KERBEROS_V4.

El paquete **Cyrus SASL** package contiene una Capa Simple de Autenticación y Seguridad, un método para añadir soporte de autenticación a protocolos basados en conexiones. Para usar **SASL**, un protocolo incluye un comando para identificar y autentificar un usuario en un servidor y, opcionalmente, negociar la protección de las subsecuentes interacciones de protocolos. Si su

uso es negociado, una capa de seguridad es insertada entre el protocolo y la conexión.
Se utiliza para permitir hacer relay. La librería **SASL** de **Cyrus** también usa la librería **OpenSSL** para cifrar los datos.

Es un framework que provee mediante plug-ins métodos de autentificación comúnmente utilizados por programas de correo como **Cyrus Imap**, Courier, **Postfix**, Sendmail, etc Permite definir nuestros propios plug-ins para hacer implementaciones propias de los métodos comúnmente usados o definir nuestros propios métodos.

2.7 SMTP (Simple Mail Transfer Protocol o Protocolo Simple de Transferencia de Correo Electrónico)

Es un protocolo estándar de Internet del Nivel de Aplicación utilizado para la transmisión de correo electrónico a través de una conexión TCP/IP. Este es de hecho el único protocolo utilizado para la transmisión de correo electrónico a través de Internet. Es un protocolo basado sobre texto y relativamente simple donde se especifican uno más destinatarios en un mensaje que es transferido.

Para determinar el servidor **SMTP** para un dominio dado, se utilizan los registros MX (Mail Exchanger) en la Zona de Autoridad correspondiente al ese mismo dominio contestado por un Servidor DNS. Después de establecerse una conexión entre el remitente (el cliente) y el destinatario (el servidor), se inicia una sesión **SMTP**, ejemplificada a continuación.

Cliente:	$ **telnet marana.isi.unaleon.edu.ni 25**
Servidor:	Trying 127.0.0.1...
	Connected to localhost.
	Escape character is '^]'.
	220 webmail.marana.isi.unanleon.edu.ni ESMTP Postfix (Ubuntu)
Cliente:	**HELO localhost**
Servidor:	250 webmail.marana.isi.unanleon.edu.ni
Cliente:	**MAIL FROM:<martha@marana.isi.unanleon.edu.ni>**
Servidor:	250 ok
Cliente:	**RCPT TO:<ana@marana.isi.unanleon.edu.ni>**
Servidor:	250 ok
Cliente:	**DATA**
Servidor:	354 End data with <CR><LF>.<CR><LF>
Cliente:	**Hola. Este es un mensaje de prueba.**

	Adiós.
	.
Servidor:	250 OK: queued as 5F496BEE9
Cliente:	**QUIT**
Servidor:	221 Bye
Servidor:	Connection closed by foreign host.

Es un protocolo simple pero inseguro, la mayoría de los **MUA** envían correos a los **MTA** Mail Transport Agent (Agente de Transporte de Correos), por medio de **SMTP**.

Entre los servidores **SMTP** tenemos: **Postfix**, Qmail, Exim, Sendmail, Microsoft Exchange. Y su puerto Web-known definidos: **SMTP** 25/tcp y **SSMTP** 465/tcp.
La descripción completa del protocolo original **STMP** está definido en el RFC 821, aunque el protocolo utilizado hoy en día, también conocido como **ESMTP** (Extended Simple Mail Transfer Protocol), está definido en el RFC 2821. **SMTP** trabaja sobre TCP en el puerto 25.

2.8 Protocolo Local de la Transferencia del Correo o LMTP

El protocolo de la transferencia del correo o el **LMTP** local es un derivado del **smtp**, el Simple Mail Transfer Protocol.

Se diseña como alternativa al smtp normal para las situaciones donde el lado de recepción no tiene una coleta del correo, tal como un agente **MDA** Mail Delivery Agent (Agente de entrega de correo) del reparto del correo que entienda conversaciones del **smtp**.

LMTP es un protocolo del nivel de la aplicación, que funciona encima de TCP/IP.

Aunque una conversación de **LMTP** utiliza los mismos comandos que una conversación del smtp.

SMTP (Simple Mail Transfer Protocol) y sus extensiones **ESMTP** (SMTP Service Extensions) proporcionan un mecanismo para transferir correo fiable y eficientemente. El diseño del protocolo **SMTP** requiere que el servidor maneje colas de envío de correo.

En ciertas circunstancias, fuera del área que engloba el intercambio entre hosts independientes en redes públicas, es deseable implementar un sistema donde

el receptor del correo no maneje colas, como es el caso de un **MDA** (Mail Delivery Agent). Esto es precisamente lo que hace el protocolo **LMTP** (Local Mail Transfer Protocol).

Aunque **LMTP** es una alternativa al protocolo **ESMTP**, usa (con algunos pequeños cambios) la sintaxis y la semántica de **ESMTP**. Este diseño permite al **LMTP** utilizar las extensiones definidas para el **ESMTP**. **LMTP** no debería ser nunca usado en el puerto 25.

2.9 Amavisd-new

Amavisd-new es una interfaz de alto rendimiento y fiabilidad entre el **MTA** o Mail Transport Agent (Agente de Transporte de Correos) y uno o más filtros de contenidos: antivirus o el módulo Mail::SpamAssassin de Perl. Está escrito en Perl, asegurando alta fiabilidad, portabilidad y facilidad de mantenimiento. Se comunica con el **MTA** vía (E) **SMTP o LMTP**, o mediante el uso de otros programas. No existen problemas de sincronización en su diseño que pudieran causar pérdidas de correos. Interfaz para línea de comando y **MTA**.

Normalmente se posiciona dentro o cerca del gestor de correo principal, no necesariamente donde se ubiquen las cuentas de correo de los usuarios (donde tiene lugar el envío final).

Cuando está habilitado el uso de Mail::SpamAssassin (SA), se llama a SA una sola vez por mensaje (independientemente del número de destinatarios). **Amavisd-new** se beneficia del uso del módulo de Perl Net::Server, el cuál ofrece un rápido entorno multihilo. **Amavisd-new** ofrece un servidor SMTP que cumple con el RFC 2821, un servidor **LMTP** que cumple con el RFC 2033, un cliente SMTP y genera notificaciones de estado de envío (o no) que cumplen los RFC 1892 y 1894. Esto lo hace adecuado para múltiples analizadores de virus y de correo publicitario en plataformas de correo donde la fiabilidad y el cumplimiento de los estándares son importantes.

2.10 SpamAssassin

SpamAssassin es un filtro de correo que trata de identificar el spam mediante el análisis del texto y el uso en tiempo real de algunas listas negras a través de Internet.

A partir de su base de datos de reglas, utiliza un amplio abanico de pruebas heurísticas en las cabeceras y el cuerpo de los correos para identificar el spam, también conocido como correo electrónico comercial no solicitado. Una vez identificado, el correo puede ser opcionalmente marcado como spam o más tarde filtrado usando el cliente de correo del usuario.

SpamAssassin normalmente identifica acertadamente entre un 95 y un 99% del spam, dependiendo del tipo de correo que se reciba.

2.11 Clam Antivirus

ClamAV es una herramienta antivirus **GPL** (General Public License o Licencia Pública General) para UNIX. El propósito principal de este software es la integración con los servidores de correo (escaneo de datos adjuntos). El paquete proporciona un servicio multihilo flexible y escalable, un analizador de línea de comandos y una utilidad para la actualización automática vía Internet. Los programas están basados en una librería distribuida con el paquete **Clam AntiVirus**, la cual puede ser usada por su propio software. Y lo más importante, la base de datos se mantiene actualizada constantemente.

Otras características destacables son el soporte de firmas digitales en la actualización de la base de datos, el análisis durante el acceso bajo Linux y FreeBSD, la detección de más de 20000 virus, gusanos y troyanos, el soporte integrado para archivos comprimidos con Rar, Zip, Gzip y Bzip2 y formatos de correo Mbox, Maildir y ficheros crudos de correo.

2.12 Mailman

Mailman es un software libre que permite gestionar listas de distribución, noticias y correo electrónicos. **Mailman** está integrado con la Web, permitiendo a sus usuarios una fácil administración de sus cuentas, así como a sus propietarios administrar las listas. **Mailman** incluye soporte para crear archivos de correos, procesamiento automático de correo rechazado, filtrado de contenido, envío en modo compendio o resumen, filtros de spam, etc.

2.13 SquirrelMail

SquirrelMail es un paquete de correo por Web basado en estándares y escrito en PHP 4. Incorpora soporte PHP para los protocolos **IMAP** y **SMTP**, y todas sus páginas se crean en puro HTML 4.0 (sin requerir el uso de JavaScript), de modo que se garantice la máxima compatibilidad entre navegadores. Tiene muy pocos requerimientos y es muy fácil de instalar y configurar. SquirrelMail tiene toda la funcionalidad que se espera de un cliente de correo electrónico, incluyendo soporte de MIME, agendas de contactos y gestión de carpetas.

Opciones Básicas de Squirrelmail

- ✓ Componer

Redactar y enviar mensajes con adjuntos.

- ✓ Direcciones

Libreta de direcciones.

- ✓ Carpetas

Permite manipular carpetas.

- ✓ Opciones

Ajustar las opciones de Squirrelmail.

- ✓ Buscar

Realizar un filtrado de los correos en base a un patrón.

Cuenta con multitud de funciones interesantes. El usuario puede ampliarlo y modificarlo fácilmente, gracias a la arquitectura de plug-ins.

Una de sus principales cualidades es su estabilidad, en lo que súpera ampliamente a otros clientes de correo. El acceso además de poder ser desde cualquier navegador y ordenador, se lleva a cabo de una forma segura (a través de **SSL**).

Las características más destacables de este webmail son:

- Gestión de carpetas.
- Internacionalización.
- Libro de direcciones personal y acceso a otros servicios de LDAP.
- Búsquedas de direcciones.
- Servicio de búsqueda en emails.
- No necesita ninguna base de datos para funcionar.
- Arquitectura de plug-ins.
- Interfaz de usuario sencilla y potente.
- Múltiples temas.
- Configuración de las vistas de mensajes: número de mensajes visibles en pantalla, campos visibles y orden.
- Comprobación de correos entrantes cada cierto intervalo de tiempo.

Los plug-ins nos aportaran:

- Autocompletado de direcciones de correo al escribir un e-mail.
- Filtros de mensajes según direcciones de correo o subject.
- Filtrado de spamming.
- Descarga de correo de múltiples cuentas POP.
- Envío de páginas HTML comprimidas.
- Utilidad de corrección de correos en cualquier idioma.
- Traducción de correos a diferentes lenguas.

- Reloj.
- Posibilidad de añadir direcciones de correo de e-mails entrantes o contenidas en un e-mail a nuestro libro de direcciones de forma automática.

2.14 Openssl

Características:

- Herramienta que permite implementar los protocolos **SSL** v2/v3 y **TLS** v1.
- Gran variedad de librerías criptográficas.
- Muy robusto y funcional.
- Permite, entre otros, generar certificados digitales y entidades certificadoras.

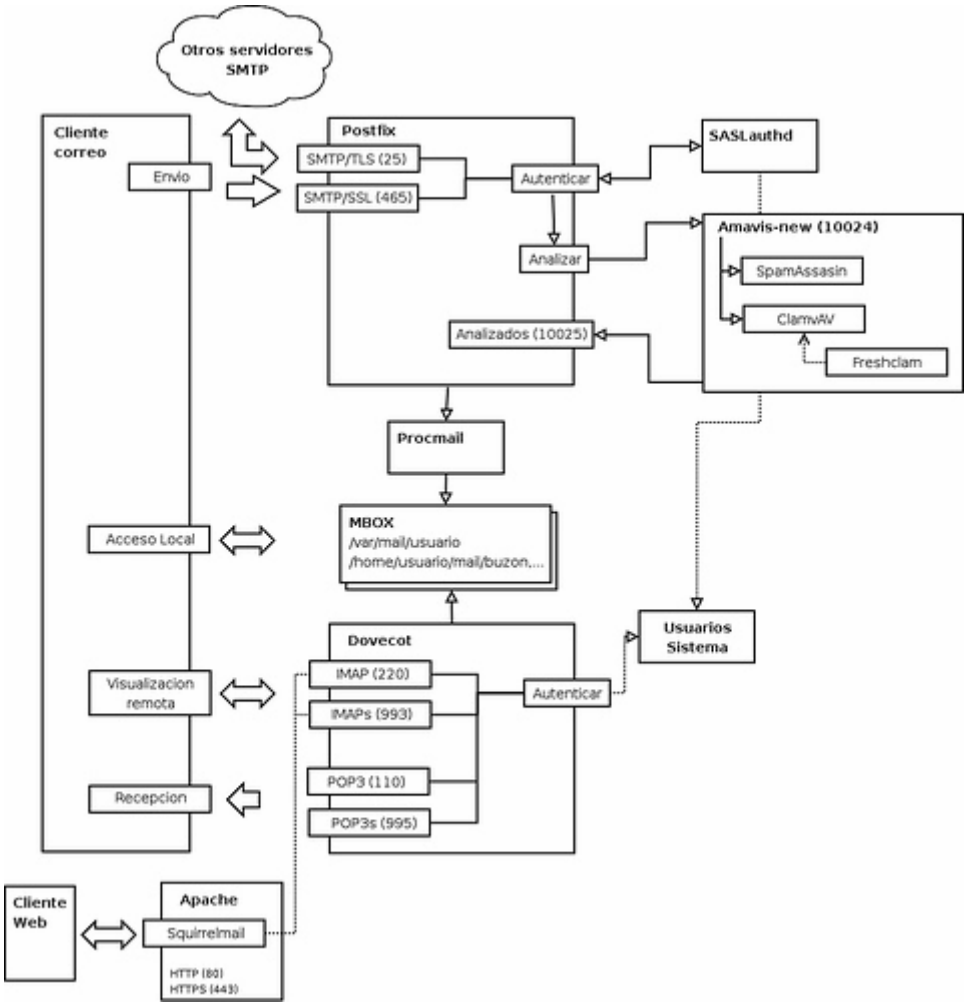
3. Instalación y configuración del correo.

Las versiones de software utilizado son:

- Linux-Ubuntu 2.6.12-9-386
- Cyrus SASL: 2.1.18-1
- Cyrus IMAP: 2.1.18-1
- Postfix: 2.1.5-9
- SpamAssassin: 3.0.3-1
- ClamAV: 0.84-2
- Amavisd-new: 20030616p10-5
- Squirrelmail: 1.4.4-5
- Mailman: 2.1.5-8

3.1 Esquema general del servidor de correo postfix y cyrus.

Esquema 2.



3.2 Instalación y configuración de Cyrus IMAP y SASL

Los paquetes necesarios para instalar **Cyrus IMAP** son cyrus21-admin, cyrus21-common, cyrus21-doc y cyrus21-imapd, y para **Cyrus SASL** son libsasl2, sasl2-bin y libsasl2-modules. Empezaremos por las librerías **SASL** y el servicio saslautd.

libsasl2
: Librería de que realiza la abstracción a los módulos de autentificación.

libsasl2-modules
: Módulos de autentificación, a través de ellos se puede autentificar contra sasldb.

sasl2-bin
: Demonio de autentificación y utilidades para el manejo de la base de datos de usuario sasldb, también crea una base de datos de usuario vacía, en formato Berkeley Database, que se encuentra en /etc/sasldb2.

Como *root*:

 # apt-get install libsasl2 sasl2-bin libsasl2-modules

NOTA: Si los repositorios de UBUNTU no se encuentran activados, activarlos de la siguiente manera:

1. Editar el archivo /etc/apt/sources.list
   Descomentar todas las opciones comentadas.
   Guardar los cambios.

2. apt-get update

Editamos el fichero ***/etc/default/saslauthd*** y modificamos el parámetro *MECHANISMS.*

```
START=yes
MECHANISMS="sasldb"
```

Iniciamos el demonio *saslauthd* con el comando:

 # /etc/init.d/saslauthd start

Una vez reiniciado el demonio saslauthd podemos añadir y borrar usuarios con el comando saslpasswd2 y con el comando sasldblistusers2 podemos listar los usuarios existentes, los parámetros más utilizados son: -c y -d, para añadir y borrar respectivamente.

saslauthd: El servidor de autenticación **SASL**.

sasldblistusers2: Se usa para listar los usuarios en la base de datos de contraseñas de **SASL**.

saslpasswd2: Se usa para establecer y borrar la contraseña de un usuario **SASL** y los mecanismos específicos secretos en la base de datos de contraseñas de **SASL**.

libsasl2.so: Librería de autenticación de propósito general para aplicaciones cliente y servidor.

Creamos el usuario Martha:
 # saslpasswd2 -c martha

Para ver si se creo el usuario tecleamos:

 # sasldblistusers2
martha@lab1pc12: userpassword

Si queremos eliminar un usuario utilizamos el comando -d:

 # saslpasswd2 -d martha
Cuando ya tenemos creado un usuario debemos asociarlo al dominio con el comando –u:

 # saslpasswd2 -c martha -u marana.isi.unanleon.edu.ni

Volvemos a listar los usuarios existentes

 # sasldblistusers2

martha@lab1pc12: userpassword

martha@marana.isi.unanleon.edu.ni: userpassword

3.3 Instalación del servidor Cyrus IMAP

 # apt-get install cyrus21-admin cyrus21-common cyrus21-doc cyrus21-imapd

Este es el programa que va a dar servicio imap. Los paquetes necesarios instalar son:

Cyrus21-common: Ficheros comunes a todos los paquetes de **Cyrus**.
Cyrus21-imapd: Servidor de correo **IMAP**.
Cyrus21-admin: Herramientas para la administración de los buzones.

Al instalar estos programas se crea una jerarquía de directorios en **/var/spool/cyrus/mai**l para almacenar los buzones de correo. Todos estos ficheros tienen como propietario a **Cyrus** y grupo mail.

También es necesario instalar el paquete **cyrus21-clients**, que proporciona las herramientas **imtest**, la cual nos será útil para comprobar el buen funcionamiento de nuestro servidor a medida que vaya avanzando la configuración y el uso de los diversos protocolos (**TLS**, **LMTP**, mecanismos **SASL**, etc).

Los dos ficheros de configuración de **Cyrus**:

/etc/cyrus.conf: En donde se configura que servicios se va a arrancar al levantar el sistema **Cyrus.**

/etc/imapd.conf: Configuracion especifica para el servicio **imap** de **cyrus**.

Comprobar si el grupo sasl tiene permisos de lectura sobre la Base de Datos de usuarios **/etc/sasldb2.**

 # ls -ls /etc/sasldb2
8 -rw-rw----- 1 root sasl 12288 2006-03-22 22:23

Comprobar que el usuario **Cyrus** pertenezca a este grupo

 # cat /etc/group
sasl:x:45:cyrus

Fichero /etc/cyrus.conf

Este fichero consta en tres partes:

START: Lista los scripts que se ejecutaran antes de que se arranquen los servicios, inicializa la base de datos y lanza los servicios de larga ejecución.
SERVICE: Esta sección es el corazón del fichero **/etc/cyrus.conf.**
EVENTS: Se usa para llevar a cabo las tareas programadas de limpieza y mantenimiento.
Dentro de los servicios se ha de tener activados (descomentado):

imap, imaps: Para dar servicio de **imap** (puerto 143) e **imap** seguro (993). Al menos uno deberá estar activado asta este momento solo es necesario tener activado el servicio **imap**.

lmtp, lmtpunix: Mediante estos protocolos se le alimentan los mensajes para que cyrus los reparta entre sus usuarios. El primero utiliza socket TCP/IP y el segundo sockets unix. Si la fuente de mensajes (postfix) están en la misma maquina se recomienda usar **lmtpunix**, en caso contrario es necesario usar **lmtp** debe activarse el que se vaya a utilizar.

En el fichero /etc/cyrus.conf
Es necesario comentar la línea donde se declara la ejecución del servicio **pop3**.
Y nos quedara un fichero como este:

```
        START {
       recover     cmd="/usr/sbin/ctl_cyrusdb −r"
       delprune   cmd="/usr/sbin/ctl_deliver −E 3"
       tlsprune    cmd="/usr/sbin/tls_prune"
    }

    SERVICES {
       imap        cmd="imapd −U 30" listen="imap" prefork=0 maxchild=100
       lmtpunix  cmd="lmtpd" listen="/var/run/cyrus/socket/lmtp" prefork=0
maxchild=20
       sieve                    cmd="timsieved"   listen="localhost:sieve"   prefork=0
maxchild=100
       notify        cmd="notifyd" listen="/var/run/cyrus/socket/notify" proto="udp"
prefork=1
   }

     EVENTS {
       checkpoint cmd="/usr/sbin/ctl_cyrusdb −c" period=30
       delprune    cmd="/usr/sbin/ctl_deliver −E 3" at=0401
       tlsprune     cmd="/usr/sbin/tls_prune" at=0401
}
```

En esta configuración sé utilizado socket Unix debido al mejor rendimiento que ofrecen y se van a ejecutar todos los servicios en la misma máquina.

Luego de haber hecho cambio en el fichero debemos reiniciamos el servidor con el comando **/etc/init.d/cyrus21 restart**.

El fichero /**etc/imapd.conf**
Es el fichero de configuración del servidor **Cyrus IMAP** y en él se definen los parámetros locales para **IMAP**.
Entre algunas de las opciones más relevantes y sus valores recomendados tenemos:

altnamespace: Esta opción viene por defecto con el valor no, forzando que las subcarpetas de usuario se creen debajo de inbox. Se usará el valor por defecto a no.

Unixhierarchysep: no. Es un separador de subcarpetas.

lmtp_downcase_rcpt: Sirve para forzar que el nombre de usuario se convierta a minúsculas, se usará el valor a yes (basta con descomentar la línea).

admins: Permite definir los usuarios que tendrán permisos de administrador sobre todos los buzones del sistema, por lo que bastará con descomentar la línea del fichero.

allowanonymouslogin: no En principio no se va a permitir usuarios anonimos.

auto createquota: 0 Si es distinto de cero se permite que los usuarios creen su INBOX y se le aplica la quota inidicada.

hashimapspool: true El directorio del spool es haced.

lmtp_admins: Si se van a usar sockets UNIX no es necesario descomentar esta opción, pues el usuario postman (valor por defecto) es autenticado automáticamente.

umask: Permite definir los permisos con los cuáles se guardarán los ficheros y subdirectorios dentro de **/var/spool/cyrus/mail**. Por defecto tiene el valor 077 (lectura y escritura para el propietario, nada para el resto), pero es conveniente permitir que el grupo (mail por defecto) tenga también permisos de lectura, pues de ese modo otras aplicaciones podrán leer el contenido de los emails, por ejemplo amavisd−new (bastará con que añadamos al usuario con el cuál se ejecutan a ese grupo). Y remplazarlo por el valor **027**.

allowplaintext: Mediante esta opción decidimos si vamos a permitir uso del mecanismo de autentificación sasl plain. Es recomendable mantener el valor por defecto **yes**.

sasl_mech_list: PLAIN Mecanismos permitidos para la autentificación sasl. Se debe descomentar esta opción.

sasl_minimum_layer: Hay que descomentar esta opción, y dejarla con un valor de 0 (valor por defecto), que permite login de texto plano.

sasl_maximum_layer: Es pertinente dejar el valor por defecto, 256 (no descomentar esta línea).

sasl_auxprop_plugin: Permite especificar los plugins del auxpropd que deseamos cargar, en el caso de estar usando sasl_pwcheck_method: auxprop. Es necesario descomentar esta línea para que use sasldb.

sasl_pwcheck_method: Esta opción queda igual con **auxprop**.

sieveusehomedir: false No leer el fichero ~/.sieve. Esto solo tiene sentido en sistemas en donde los usuarios del correo son usuarios del sistema.

sievedir: /var/spool/sieve Es el directorio en donde se buscan los scripts sieve.

tls_ca_path: /etc/ssl/certs Directorio en donde están los certificados de las autoridades certificadoras.

tls_session_timeout: 1440 Timeout de la sesion.

tls_cipher_list:TLSv1:SSLv3:SSLv2:!NULL:!EXPORT:!DES:!LOW:@STRENGTH. Cifrados que se permiten.

lmtpsocket: /var/run/cyrus/socket/lmtp Socket a utilizar para recibir mensajes. Debe ser el mismo que aparece en el fichero de configuracion **/etc/cyrus.conf**.

idlesocket: /var/run/cyrus/socket/idle Socket a utilizar por el programa idled encargado de revisar cambios en los mailboxes. Debe ser el mismo que aparece en el fichero de configuracion /etc/cyrus.conf.

El **fichero /etc/impad.conf** quedara como se presenta a continuación:

```
 configdirectory:  /var/lib/cyrus
defaultpartition: default
partition−default:  /var/spool/cyrus/mail
partition−news:  /var/spool/cyrus/news
newsspool:  /var/spool/news
altnamespace: no
unixhierarchysep: no
lmtp_downcase_rcpt: yes
admins: cyrus
allowanonymouslogin: no
popminpoll: 1
autocreatequota: 0
umask: 027
sieveusehomedir: false
sievedir:  /var/spool/sieve
hashimapspool: true
allowplaintext: yes
sasl_mech_list: PLAIN
sasl_minimum_layer: 0
sasl_pwcheck_method: auxprop
sasl_auxprop_plugin: sasldb
sasl_auto_transition: no
tls_ca_path: /etc/ssl/certs
tls_session_timeout: 1440
tls_cipher_list:
TLSv1:SSLv3:SSLv2:!NULL:!EXPORT:!DES:!LOW:@STRENGTH
lmtpsocket:  /var/run/cyrus/socket/lmtp
idlesocket:  /var/run/cyrus/socket/idle
notifysocket:  /var/run/cyrus/socket/notify
```

Cada vez que se realizamos un cambio en el fichero **/etc/imapd.conf** es necesario indicarle al servidor **Cyrus** que relea su contenido.

/etc/init.d/cyrus21 restart o /etc/init.d/cyrus21 reload

3.3.1 Configuración de buzones de Correo

Creamos el usuario **Cyrus**:
 # saslpasswd2 -c cyrus

En nuestra base de datos de usuario debería contener los usuarios: martha y cyrus, para comprobar que dichos usuarios existen los listamos.

sasldblistusers2
Y nos dará como resultado:
martha@lab1pc12: userPassword

cyrus@lab1pc12: userPassword

martha@marana.isi.unanleon.edu.ni: userPassword

Tabla 2. Comandos para la administración del programa cyradm

comando	***Alias***	***Función***	***sintaxis***	***Ejemplos***
createmailbox	Cm	crear buzones de correo	cm<buzon>	cm user.matha
deletemailbox	Dm	Borrar buzones de correo	dm<buzon>	dm user.martha
listacl	Lam	listar las ACL de un buzon	lam<buzon>	lam user.martha
setacl	Sam	Establecer las ACL en un buzón	sam<buzon><usuario><permisos>	sam user.martha martha lrs
deleteacl	Dam	Borrar las ACL de un buzón	dam<buzon><usuario>	dam user.martha martha

Creación de Buzones de Prueba

cyradm –user cyrus localhost
IMAP Password:
localhost.localdomain> **cm user.martha**
localhost.localdomain> **lam user.martha**
martha lrswipcda
localhost.localdomain>**exit**

Cuando ya tenemos creado los un buzón a través de la herramienta **imtest** comprobamos el correcto funcionamiento del servidor.
Como root:
 # imtest –a martha –w costaazul –m login localhost

Resultado del imtest:

```
root@lab1pc12:/home/usuario# imtest -a martha -w costaazul -m login localhost
S: * OK lab1pc12 Cyrus IMAP4 v2.1.18-IPv6-Debian-2.1.18-1ubuntu1 server ready
C: C01 CAPABILITY
S: * CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ MAILBOX-REFERRALS NAMESPACE UIDPLUS ID
MULTIAPPEND SORT THREAD=ORDEREDSUBJECT THREAD=REFERENCES IDLE ANNOTATEMORE
S: C01 OK Completed
C: L01 LOGIN martha {9}
S: + go ahead
C: <omitted>
S: L01 OK User logged in
Authenticated.
Security strength factor: 0
```

Pulsamos Ctrl+C para abandonar el programa de prueba.

```
C: Q01 LOGOUT
Connection closed.
```

4. Instalación y configuración de postfix.

Debemos ejecutar el siguiente comando como root:
apt-get install postfix postfix-tls postfix-doc postfix-pcre mime-codecs

- **postfix:** Este es el paquete principal de **Postfix**.
- **postfix-doc:** Contiene la Documentación.
- **postfix-pcre:** Soporte de expresiones regulares.
- **postfix-tls:** Soporte **TLS** y **SASL** (**SMTP** autentificado).

Esto nos dejará instalados en el sistema todo lo necesario para la configuración posterior de **Postfix**. En el caso de que no tuviésemos el paquete **postfix** anteriormente, el script de postinstalación de este paquete nos mostrará unas pantallas que permiten configurar de un modo básico el servidor. Si ya estaba instalado el paquete, entonces deberemos llamar a ese asistente de modo manual mediante el comando:

dpkg-reconfigure postfix

Cuando reconfiguramos postfix nos aparecerá las siguientes preguntas:

1. **General type of configuration?** Internet Site
2. **Where should mail for root go?** martha
3. **Mail name?** marana.isi.unanleon.edu.ni

4. **Other destinations to accept mail for?** marana.isi.unanleon.edu.ni,lab1pc12,webmail.marana.isi.unanleon.edu.ni,localhost.localdomain,localhost.localdomain, localhost
5. **Force synchronous updates on mail queue?** No
6. **Local networks?** 127.0.0.0/8
7. **Use procmail for local delivery?** No
8. **Mailbox size limit?** 0
9. **Local address extensión character?** +

El modo *internet site* se caracteriza porque el propio servidor se encarga de repartir los mensajes a sus destinatarios directamente, sin pasar por otro servidor predefinido. Para usar este modo, en el fichero de configuración **/etc/postfix/main.cf** no debe estar definida la opción **relayhost**.

Fichero /etc/postfix/master.cf

Hay que puntualizar que ubuntu no ejecuta el programa de postfix que maneja el protocolo lmtp en un chroot, habría que ser accesible al socket **/etc/postfix/master.cf:**

En caso de que se quiera ejecutar en un chroot habrá que hacer accesible el socket **/var/run/cyrus/socket/lmtp** desde su jaula, bien haciendo un enlace duro, bien modificando el path del socket en la configuracion de **cyrus**.

Debemos asegurarnos que en este fichero contenga la siguiente línea:

```
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
#
==========================================================
lmtp        unix    -       -       n       -       -       lmtp
```

De esta forma el programa no se ejecutara en un chroot (en una jaula).

El fichero /etc/postfix/main.cf
Postfix 2.x no pasa a minúsculas los destinatarios en las entregas por **LMTP**, es aconsejable usar la opción ***lmtp_downcase_rcpt: yes*** en el fichero ***/etc/imapd.conf***. Para sockets Unix, el transporte de Postfix se especifica como ***lmtp:unix:/var/run/cyrus/socket/lmtp***.

lmtp: Este es el método a utilizar si se quiere aprovechar las ventajas de sieve, ya que mediante este método se aplican los filtros sieve. Los métodos

anteriores no utilizan sieve porque se basan en el programa cyrdeliver, que hace la entrega directa en el buzón indicado.

Es necesario también un servicio lmtpd de **Cyrus** escuchando en ese socket, por lo tanto es necesario que exista una línea siguiente:

```
lmtpunix cmd="lmtpd" listen="/var/run/cyrus/socket/lmtp" prefork=0 maxchild=20
```

En la sección *SERVICES* del fichero ***/etc/cyrus.conf*.**

dpkg-statoverride para asegurarse que la configuración de los permisos del socket no es sobrescrita por los paquetes de **Cyrus**. Recuerde que **Postfix** ejecuta el transporte **LMTP** con el usuario definido en ***/etc/postfix/master.cf***, por defecto **postfix**.

- Crear un grupo llamado *lmtp*:
  # **addgroup lmtp**

- Agregar el usuario *postfix* a ese grupo:
  # **adduser postfix lmtp**

- Corrija los permisos del directorio del socket:
  # **dpkg-statoverride --force --update --add cyrus lmtp 750 /var/run/cyrus/socket**

- Reinicie Postfix y Cyrus:
  # **/etc/init.d/postfix restart**
  # **/etc/init.d/cyrus21 restart**

El fichero no quedara de esta manera /**etc/postfix/main.cf** como éste:

```
setgid_group = postdrop
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no
append_dot_mydomain = no
delay_warning_time = 4h
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
program_directory = /usr/lib/postfix
myhostname = webmail.marana.isi.unanleon.edu.ni
```

```
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, $mydomain, localhost.$mydomain,
lab1pc12, webmail.marana.isi.unanleon.edu.ni localhost
myorigin = $mydomain
mynetworks = 127.0.0.0/8
mailbox_size_limit = 0
recipient_delimiter = +
local_recipient_maps =

mailbox_transport = lmtp:unix:/var/run/cyrus/socket/lmtp
```

En el fichero ***/etc/mailname*** colocamos:

```
marana.isi.unanleon.edu.ni
```

En el fichero ***/etc/hostname*** es:

```
Lab1pc12
```

Donde ***/etc/hosts*** contiene:

```
127.0.0.0 localhost.localdomain localhost lab1pc12
127.16.0.240 lab1pc12 marana.isi.unanleon.edu.ni marana.isi.unanleon.edu.ni
webmail marana.isi.unanleon.edu.ni lab1pc12
```

Y donde ***/etc/resolv.conf*** contiene:

```
nameserver 172.16.0.1

nameserver 192.107.104.01

nameserver 127.0.0.1
```

Comprobamos el buen funcionamiento, primero nos aseguramos que los servicios estén escuchando en los puertos correspondientes:

 # netstat -an|grep LISTEN

```
tcp        0      0 0.0.0.0:143             0.0.0.0:*       LISTEN
tcp        0      0 127.0.0.1:2000          0.0.0.0:*       LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*       LISTEN
```

El puerto 25 es de **smtp**.

El 143 es de **imap**.

El 2000 de **sieve**.

Luego podemos mandar un correo electrónico a un buzón local desde de la propia máquina local o desde una exterior. De martha a ana.

Nota: Tenemos que crear al usuario ana y crear su buzón correspondiente.

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 localhost.localdomain ESMTP Postfix (Ubuntu)
HELO localhost
250 localhost.localdomain
MAIL FROM: <martha@marana.isi.unanleon.edu.ni>
250 Ok
RCPT TO: <ana@marana.isi.unanleon.edu.ni>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
hola ana como estas estamos en señal de prueba.
.
250 Ok: queued as 7E45A68030
QUIT
221 Bye
Connection closed by foreign host.
```

Ya podemos ver que nuestro correo ha sido enviado al buzón ana@marana.isi.unanleon.edu.ni ejecutando el comando:

```
# ls /var/spool/cyrus/mail/a/user/ana/
1.  cyrus.cache  cyrus.header  cyrus.index
```

Para revisar los correos debemos configurar un visor de correo en este caso utilizamos el evolution. (Ver sección de anexos)

Podemos añadir el parámetro *-v* para ver un log del servidor de correo mucho más detallado.

 # *tail -f /var/log/mail.log | colorize*

Instalamos el paquete *colorize:*

 # apt-get install colorize

La línea quedaría así en el fichero **/etc/postfix/master.cf**

```
smtp      inet  n       -       -       -       -       smtpd –v
```

Esta línea es la que le indica a **postfix** que lea el correo de entrada por el puerto smtp (25) y que el demonio smtpd se encargue de él.

Procedamos a activar **SASL** sobre **postfix**. Autentificaremos a los clientes conectados mediante *smtp* para que puedan hacer *relay* sobre el servidor de correo. Para ello modificaremos la opción smtpd_recipient_restrictions del **fichero /etc/postfix/main.cf**:

```
smtp_sasl_auth_enable = no
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = lab1pc12
smtpd_recipient_restrictions =
   permit_mynetworks,
   permit_sasl_authenticated,
   reject_unauth_destination
smtpd_sasl_security_options = noanonymous
```

Así se permite hacer relay a los clientes sin autentificar que pertenezcan a las redes indicadas en mynetworks y a los clientes autentificados mediante el método sasl. Se rechazaran los que están sin autentificar. Además es necesario utilizar **tls** para autentificarse. Con la opción **smtpd** sasl local domain=lab1pc12 se indica que compruebe el usuario entrando en /etc/sasldb2 con usuario@lab1pc12.

Una vez activado el uso de **SASL** en **postfix** tendremos que indicar seguidamente el mecanismo a usar a través de **SASL** y para ello procedemos a editar el fichero **/etc/postfix/sasl/smtpd.conf**

```
pwcheck_method: saslauthd

mech_list : plain login
```

Como postfix se ejecuta por defecto en un chroot (jaula) localizada en **/var/spool/postfix** no puede acceder al socket y demás ficheros de autenticación SASL localizados **en /var/run/saslauthd** por lo que postfix tratara de encontrarlos en **/var/spool/postfix/var/run/sasl/authd**, por lo cual se debe realizar algunos cambios en el fichero **/etc/postfix/master.cf**, en la línea del smtp.

```
#==========================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
#==========================================================
smtp      inet  n       -       n       -       -       smtpd
```

Añadir al usuario postfix al grupo **SASL**
adduser postfix sasl

Y después listar para ver si se añadió al usuario **postfix**
less /etc/group | grep sasl

Reiniciar el servidor de autenticación:
/etc/init.d/saslauthd restart

Reiniciar el servidor de correo:
/etc/init.d/postfix restart.

Generar la cadena alfanumérica que nos permitirá la autentificación durante el proceso de comunicación.

perl -MMIME::Base64 -e 'print encode_base64("martha\0martha\0costaazul");'

Para la combinación martha/martha la cadena alfanumérica es:
bWFydGhhAGlhcnRoYQBjb3N0YWF6dWw=

Ahora comprobamos el buen funcionamiento utilizando autentificación PLAIN y la cadena alfanumérica que obtuvimos en negrita lo que escribimos.

telnet localhost 25

Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
220 localhost.localdomain ESMTP Postfix (Ubuntu)
EHLO localhost
250-localhost.localdomain
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH PLAIN
250 8BITMIME
AUTH PLAIN bWFydGhhAGlhcnRoYQBjb3N0YWF6dWw=
235 Authentication successful
QUIT
221 Bye
Connection closed by foreign host.

5. Cifrado del canal de comunicación usando TLS (Transport Layer Security o Seguridad para Capa de Transporte)

Habitualmente las comunicaciones por el protocolo **SMTP** se realizan en claro, por lo que cualquier transmisión de información confidencias es susceptible de ser interceptada. **Postfix** soluciona esto de un modo elegante, usando certificados X.509. El protocolo **TLS** está basado en **SSL** y su definición se encuentra en el RFC-2246.

La integración del protocolo **TLS** y **SMTP** se define en el RFC-2487 y se implementa en **ESMTP**, en concreto, la negociación inicial (EHLO).

El servidor ofrece **TLS** mediante la opción STARTTLS, invitando al cliente a pasar a un estado de comunicaciones cifradas.

Puede averiguar si su servidor soporta **TLS** haciendo un telnet al puerto 25 e identificándose mediante el comando EHLO. El servidor devolverá una lista de servicios, entre los cuales debe estar STARTTTLS.

5.1 Instalación del paquete openssl

apt-get install openssl

Generación de los Certificados.

1. Crear una nueva autoridad certificadora: /usr/lib/ssl/misc/CA.pl -newca
2. Realizar la petición de un certificado: /usr/lib/ssl/misc/CA.pl -newreq-nodes
3. Firmar el certificado: /usr/lib/ssl/misc/CA.pl -sign

Con la salvedad de que no debemos añadir una palabra de paso al certificado para que el servidor no se quede bloqueado esperándola al iniciarse. Luego deberemos copiar tres de los ficheros resultantes del proceso al subdirectorio /etc/postfix/ssl/:

- **cacert.pem:** El certificado de la autoridad certificadora, al cual se remitirá al cliente cuando quiera comprobar la autenticidad del certificado que le ha enviado nuestro servidor **Postfix**.

- **newcert.pem:** El certificado público que enviaremos al cliente para establecer la comunicación segura.

- **newreq.pem:** el certificado privado que almacenaremos en el servidor y del cual la parte realmente importante es la clave, que debe permanecer secreta.

Ejecutaremos entonces los siguientes comandos:

 # mkdir /etc/postfix/ssl
 # cp demoCA/cacert.pem /etc/postfix/ssl/
 # cp newcert.pem /etc/postfix/ssl/
 # cp newreq.pem /etc/postfix/ssl/
 # chown root /etc/postfix/ssl/newreq.pem
 # chmod 400 /etc/postfix/ssl/newreq.pem

Modificaciones en Postfix

Tan sólo es necesario editar el fichero **/etc/postfix/main.cf** y agregar las siguientes líneas:

```
smtpd_use_tls = yes
# smtpd_tls_auth_only = yes
smtpd_tls_key_file = /etc/postfix/ssl/newreq.pem
 smtpd_tls_cert_file = /etc/postfix/ssl/newcert.pem
 smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
 smtpd_tls_loglevel = 3
 smtpd_tls_received_header = yes
 smtpd_tls_session_cache_timeout = 3600s
 tls_random_source = dev:/dev/urandom
```

De este modo restringimos que las autenticaciones al envío se puedan hacer únicamente usando **TLS** (esta línea aparece comentada de momento). Además, obligamos a que todas las comunicaciones con el demonio smtpd se hagan a través de **TLS**. Tras reiniciar el servidor **Postfix**, ya podemos comprobar el correcto funcionamiento del mismo:

 # /etc/init.d/postfix restart

Nota: En negrita lo que tecleamos.

 # telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 webmail.marana.isi.unanleon.edu.ni ESMTP Postfix (Ubuntu/GNU)
EHLO localhost
250-webmail.marana.isi.unanleon.edu.ni
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
STARTTLS
220 Ready to start TLS
QUIT
QUIT
Connection closed by foreign host.

Entonces, si queremos asegurarnos de que las autenticaciones se realicen únicamente sobre un canal cifrado mediante el protocolo **TLS**, debemos descomentar la línea que reza **smtpd_tls_auth_only =yes**. Tras reiniciar el servidor, podemos observar las diferencias:

 # telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 webmail.marana.isi.unanleon.edu.ni ESMTP Postfix (Ubuntu/GNU)
EHLO localhost
250-webmail.marana.isi.unanleon.edu.ni
250-PIPELINING
250-SIZE 10240000
250-VRFY

250-ETRN
250-STARTTLS
250 8BITMIME
STARTTLS
220 Ready to start TLS
QUIT
QUIT
Connection closed by foreign host.

Nótese que el servidor no ofrece los métodos de autenticación antes de que se haya establecido el canal seguro. Asimismo, una vez que nos hemos cerciorado de que todo funciona adecuadamente, podemos reducir el nivel de logging cambiando el valor **3** por **1** en la opción **smtpd_tls_loglevel**.

Modificaciones en Cyrus IMAP (Internet Message Access Protocol).

Para activar el cifrado del canal de comunicaciones en **Cyrus IMAP** deberemos modificar dos ficheros, **/etc/cyrus.conf y /etc/imapd.conf**. En el primero añadiremos el servicio imaps a la lista de los que deben iniciarse con el sistema Cyrus, mientras que en el segundo configuraremos varios parámetros relacionados con **TLS** y la localización de los certificados en el disco duro.

Entonces, en la sección **SERVICES** del fichero **/etc/cyrus.conf** tan sólo debemos descomentar una línea:

```
imaps               cmd="imapd -s -U  30"  listen="imaps"  prefork=0
maxchild=100
```

Y cambiar la que antes teníamos activa para soportar **IMAP** sin **SSL** solo a través de la interfaz localhost (necesario para usar cyradm), de modo que forcemos al usuario a usar un canal cifrado:

```
imap                cmd="imapd -U  30"  listen="localhost:imap"  prefork=0
maxchild=100
```

Las líneas a descomentar o modificar del fichero **/etc/imapd.conf** son las siguientes:

```
tls_cert_file: /etc/ssl/certs/cyrus-global.pem

tls_key_file: /var/imap/cyrus-global.key

tls_ca_file: /etc/ssl/certs/cyrus-imapd-ca.pem

tls_ca_path: /etc/ssl/certs

tls_session_timeout: 1440

tls_cipher_list:
TLSv1:SSLv3:SSLv2:!NULL:!EXPORT:!DES:!LOW:@STRENGTH
```

Puede apreciarse que la parte privada del certificado se encuentra en un directorio específicamente creado para tal propósito, **/var/imap.** Esto es debido a que el usuario cyrus debe tener permisos de lectura sobre él, mientras que su ubicación estándar, que sería **/etc/ssl/private** no nos permite esa posibilidad.

Se ha usado el mismo certificado para **Postfix** que para **Cyrus**, siendo necesario ejecutar los siguientes comandos:

 # mkdir /var/imap

 # chown cyrus:mail /var/imap

 # chmod 750 /var/imap

 # cp /etc/postfix/ssl/newcert.pem /etc/ssl/certs/cyrus-global.pem

 # cp /etc/postfix/ssl/cacert.pem /etc/ssl/certs/cyrus-imapd-ca.pem

 # cp /etc/postfix/ssl/newreq.pem /var/imap/cyrus-global.key

 # chown cyrus:mail /var/imap/cyrus-global.key

 # chmod 600 /var/imap/cyrus-global.key

Téngase en cuenta que, tal y como hemos configurado el servidor **Postfix** (**smtpd**), si al enviar un correo lo hacemos a un servidor que soporta (ofrece) **TLS** sobre el protocolo **smtp**, nuestro servidor tratará de usar cifrado sobre el

canal. Por lo tanto, es muy conveniente, por no decir imprescindible, tener un certificado firmado por una autoridad certificadora reconocida.

Tras reiniciar el servidor **Cyrus (/etc/init.d/cyrus21 restart)** podemos comprobar que los cambios han surgido efecto:

 # imtest -a martha -w costaazul -m login -s localhost

Resultado del imtest:

verify error:num=20:unable to get local issuer certificate

verify error:num=21:unable to verify the first certificate

TLS connection established: TLSv1 with cipher AES256-SHA (256/256 bits)

S: * OK genma Cyrus IMAP4 v2.1.16-IPv6-Debian-2.1.16-6 server ready

C: C01 CAPABILITY

S: * CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ MAILBOX-REFERRALS NAMESPACE UIDPLUS ID NO_ATOMIC_RENAME UNSELECT CHILDREN MULTIAPPEND SORT THREAD=ORDEREDSUBJECT THREAD=REFERENCES IDLE AUTH=LOGIN AUTH=PLAIN LISTEXT LIST-SUBSCRIBED ANNOTATEMORE

S: C01 OK Completed

C: L01 LOGIN martha {8}

S: + go ahead

C: <omitted>

S: L01 OK User logged in

Authenticated.

Security strength factor: 256

Pulsando Ctrl+C abandonaremos el programa de pruebas:

C: Q01 LOGOUT

Connection closed.

6. Correo a través de Web con SquirrelMail.

6.1 Instalación de squirrelmail

Esta instalación del software se hace como root:

apt-get install squirrelmail

/usr/sbin/squirrelmail-configure.

De las opciones que nos presenta a lo largo y ancho de los menús, estas son las que personalmente hemos considerado que eran merecedoras de modificación:

```
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books (LDAP)
7. Message of the Day (MOTD)
8. Plugins
9. Database
D.  Set pre-defined settings for specific IMAP servers
C.  Turn color on
S   Save data
Q   Quit
Command >>
```

Organization Preferences: Organization Name: Bienvenidos al webmail de marana

Y guardar con la opción:

```
Command >>S
```

Y salir con:

```
Command >>Q
```

Reiniciamos apache2:
 # /etc/init.d/apache2 restart

Creamos un enlace débil en /etc/apache2/conf.d/ así:

 # ln -s /etc/squirrelmail/apache.conf /etc/apache2/conf.d/squirrelmail.conf

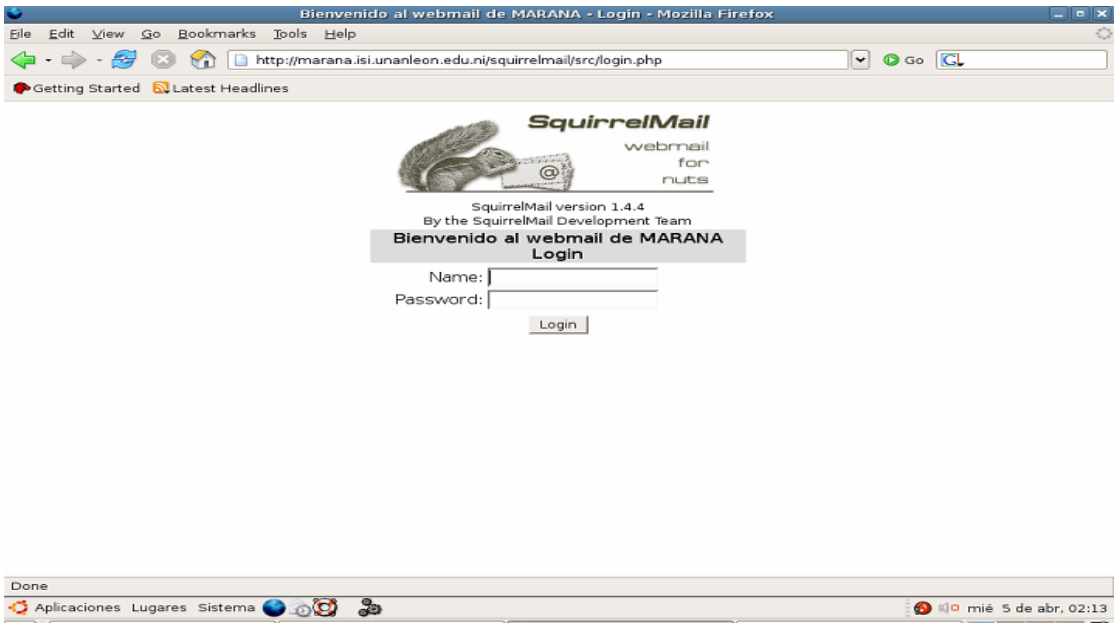
En el fichero **/etc/squirrelmail/apache.conf** descomentamos la parte del **VirtualHost** y agregamos nuestro **ServerName**.

```
<VirtualHost 1.2.3.4>
DocumentRoot /usr/share/squirrelmail
ServerName webmail.marana.isi.unanleon.edu.ni
</VirtualHost>
```

Reiniciamos el servidor Web:

 # apache2ctl graceful

Ahora ya podemos acceder mediante URL:
http://marana.isi.unanleon.edu.ni/squirrelmail



7. Filtros de contenidos.

SpamAssassin y **ClamAV** son dos filtros de contenidos que serán utilizados desde **Postfix** a través de **Amavisd-new**. Instalaremos y configuraremos primero estos filtros y después la interfaz **Amavisd-new**.

7.1 SpamAssassin

Filtrando Spam desde Postfix

La operación de un servidor de correo sobre Internet, implica la responsabilidad de no permitir el open relay, con el cual los Spammers puedan utilizar nuestros recursos para realizar sus operaciones. Un servidor de open relay es un sistema de correo que permite él envió de mensajes de sistemas externos a otros sistemas externos por medio de nuestro servidor.

Detección de Spam

Después de resguardar nuestro servidor, lo próximo a realizar es reducir la cantidad de Spam recibido por nuestros usuarios. Debemos elegir un método eficiente que permita eliminar el correo Spam. Sin embargo la existencia de falsos positivos hará difícil nuestra decisión, debido a que la mala clasificación del correo deseado por nuestros usuarios, es un problema común en la clasificación de correo Spam.

Existen dos métodos principales para la detección de Spam: la identificación del origen del Spam y mediante el análisis del contenido del mensaje en busca de frases que revelen características clásicas o no del Spam.

Detección basada en información del origen

Utilizando información como direcciones IP, nombres de host así como direcciones de correo suministradas por los clientes cuando ellos envían algún mensaje, esta técnica compara estas piezas de información con datos conocidos sobre sistemas utilizados para él envió de Spam. Sin embargo la utilización de recursos ajenos a los Spammers para él envió de Spam, como puede ser la utilización de Open Relays, puede retrasar el proceso de investigación del verdadero origen del correo basura. Además, que toda la información obtenida mediante los encabezados de algún mensaje, puede ser generalmente, fácilmente alterada por los Spammers.

Listas negras de DNS

En un esfuerzo por detener el Spam, han sido desarrollados varios servicios Anti-Spam generalmente denominados listas negras basadas en DNS o listas negras en tiempo real. Estos servicios mantienen bases de datos en las cuales se encuentran registrados host que son conocidos como open relays, o que han sido utilizados para él envió de Spam. Algunos de los ataques más comunes en contra de estos equipos, es la instalación de un proxy propio del spammer el cual no solo podría ser utilizado para enviar Spam, si no también para generar ataques de denegación de servicio.

Detección basada en contenido

Además de identificar clientes que permiten él envió de Spam desde sus equipos, también se puede detectar Spam por medio de su contenido. Ciertas cadenas de texto marcan significativamente el nivel de posibilidad de que un correo sea Spam o no, por ejemplo "Make more Money". Sin embargo este tipo de análisis puede llegar a ser muy problemático, debido a los falsos positivos que puede generar una mala clasificación del Spam.

Configuración Anti-Spam de Postfix

Existen 4 diferentes formas de detección de Spam en **Postfix**:

Reglas de detección de clientes Spam:

Reglas que trabajan con la identidad del cliente. Cada regla es una lista asignada de una o más restricciones que pueden aceptar o rechazar un mensaje. Por ejemplo, podemos incluir una regla que rechace mensajes de una dirección IP en particular.

Parámetros de verificación de sintaxis:

Son parámetros que verifican el apego a los estándares actuales en la composición del mensaje. Debido a que varios Spammers no siguen al pie de la letra los estándares que rigen el correo electrónico, nuestro sistema puede estar configurado para rechazar mensajes que vienen de sistemas mal o pobremente configurados.

Verificación de contenido:

Se puede realizar una verificación de las cabeceras y del cuerpo del mensaje por medio de expresiones regulares, las cuales nos indicarán un mensaje que probablemente sea Spam.

Clases de restricción:
Se pueden definir reglas complejas para la detección de clientes Spam por medio de clases de restricción. Estas permiten combinar restricciones en grupos para formar nuevas restricciones.

Cuando se configura **Postfix** con características Anti-Spam, también se puede especificar qué hacer con los mensajes clasificados como Spam. En general, Postfix puede rechazar los mensajes, separarlos en una cola diferente, o entregarlos a un filtro externo.

7.1.1 Instalación de SpamAssassin.

Ejecutaremos el siguiente comando como **root:**

 # apt-get install spamassassin spamc

Entrar al fichero **/etc/default/spamassassin** y lo dejamos así:

```
ENABLED=1

OPTIONS = "-c –m 10 –a -H"
```

Debido a que se usará **SpamAssassin** a través de **Amavisd-new**, éste será llamado a través del módulo de Perl Mail::SpamAssassin, manteniendo Perl el motor de reglas siempre cargado en memoria y consiguiendo la misma eficiencia que con el demonio.

El motivo por el cuál se estableció la máscara de los ficheros y subdirectorios de **Cyrus** a 027, de modo que, si añadimos el usuario amavis al grupo mail –adduser amavis, se podrán leer esos mails considerados ham o spam. Asimismo, es posible que los directorios y ficheros que se crearon por los scripts de instalación antes de realizar el cambio en ***/etc/imapd.conf*** deban ver sus permisos modificados.

7.2 Clam Antivirus

Clam Antivirus es un juego de herramientas antivirus para UNIX, liberado según licencia **GPL**, cuyo propósito fundamental es la integración con servidores de correo para el escaneo de ficheros adjuntos. Se le dan más aplicaciones, pero **ClamAV** está pensado para eso. El demonio es multihilo, el escáner es de línea de comandos y contempla una herramienta de actualización automática vía Internet. La base de datos de firmas está actualizada siempre, si bien tampoco son las más eficientes ni las más

tempranas en modificarse. Aún así, los resultados en servidores son más que satisfactorios.

7.2.1 Instalación de Clam Antivirus

Tenemos que tener el repositorio non-free en nuestro **/etc/apt/source.list**.

 # apt-get install unrar lha arj unzoo zip unzip bzip2 gzip cpio file lzop

 # apt-get install clamav clamav-base clamav-daemon clamav-freshclam libclamav1

También es necesario instalar una versión más reciente de lsb-base en nuestro caso instalamos lsb-base3.0.12.

El sitio de descarga de este paquete es:

http://ni.archive.ubuntu.com/ubuntu/spool/mail/l/lsb

Lo descargamos y lo guardamos en **/home/usuario**

 # dpkg-i /home/usuario lsb-base3.0.12

Reconfiguramos el clamav-freshclam

 # dpkg-reconfigure clamav-freshclam

Como método de actualización de la lista y los patrones de los virus del *clamav-freshclam* recomiendo seleccionar ***daemon*** y como **mirror** para la descarga el más cercano a la localización geográfica de nuestro servidor en nuestro caso **Nicaragua**. En la pregunta *Number of freshclam updates per day*, con **12** tenemos actualizada cada dos horas nuestra lista de virus. Y contestamos que **sí** a al pregunta de
Should clamd be notified after updates?. No es preciso modificar ningún fichero de configuración, pues la instalación ya nos deja todo lo que necesitamos funcionando adecuadamente.

Es necesario añadir al usuario clamav al grupo amavis:
 # adduser clamav amavis

7.3 Amavisd-new

Es una interfaz entre el **MTA** y algún verificador de contenido como scanners de virus y/o SpamAssassin, el cual nos ayuda a identificar el posible spam que recibe nuestro servidor.

Es necesario instalar como root:

apt-get install amavis-new

A continuación se presentan las líneas del **fichero /etc/amavis/amavisd.conf** que necesitan ser modificadas, en el formato definitivo (es decir, con las modificaciones ya realizadas):

```
$mydomain = 'marana.isi.unanleon.edu.ni';

$myhostname = 'webmail.marana.isi.unanleon.edu.ni';

# @bypass_spam_checks_acl = qw( . );

$final_spam_destiny = D_PASS;

$warnbannedsender = 1;

$warnbadhsender = 1;

# $virus_quarantine_to = 'virus-quarantine';

$virus_quarantine_to = "virus-quarantine\@$mydomain";

# $sa_spam_subject_tag = '***SPAM*** '; #

$banned_filename_re = new_RE(

# qr'^UNDECIPHERABLE$',

   qr'\.[^.]*\.(exe|vbs|pif|scr|bat|cmd|com|dll)$'i,

    qr'[{}]',

# qr'.\.(exe|vbs|pif|scr|bat|cmd|com)$'i,
```

```
qr'.\.(ade|adp|bas|bat|chm|cmd|com|cpl|crt|exe|hlp|hta|inf| ins|isp|js|
            jse|lnk|mdb|mde|msc|msi|msp|mst|pcd|pif|reg|scr|sct|shs|shb|vb|
            vbe|vbs|wsc|wsf|wsh)$'ix,
    qr'.\.(mim|b64|bhx|hqx|xxe|uu|uue)$'i,
# qr'^\.(zip|lha|tnef|cab)$'i,
  qr'^\.exe$'i,
  qr'^application/x-msdownload$'i,
  qr'^application/x-msdos-program$'i,
  qr'^message/partial$'i, qr'^message/external-body$'i,
 );
```

Las modificaciones realizadas al fichero dejan una configuración específica. Acto seguido se resume los porqués de los cambios realizados:

- **bypass_spam_checks_acl:** Comentamos esta línea para que ***Amavisd-new*** use ***SpamAssassin*** (por defecto viene deshabilitado su uso).

- **final_spam_destiny:** Dejamos pasar los correos identificados como *spam*, aunque siguen siendo marcados como tales mediante cabeceras en el correo. De este modo, los destinatarios seguirán recibiendo toda su correspondencia pero podrán filtrarla fácilmente usando *Sieve* y las cabeceras que ***Amavisd-new*** habrá añadido al mensaje.

- **warnbannedsender:** Activamos el envío de un mensaje de aviso al remitente de un mensaje que contuviera algún fichero adjunto con una de las extensiones prohibidas que más abajo se detallan.

- **warnbadhsender:** Igual que el anterior para ficheros con cabeceras malformadas.

- **virus_quarantine_to:** Activamos la cuarentena de los correos con virus. De este modo, cualquier correo que contenga un virus detectado será redirigido a la cuenta especificada. Así, podremos revisarlos y decidir qué hacer con ellos.

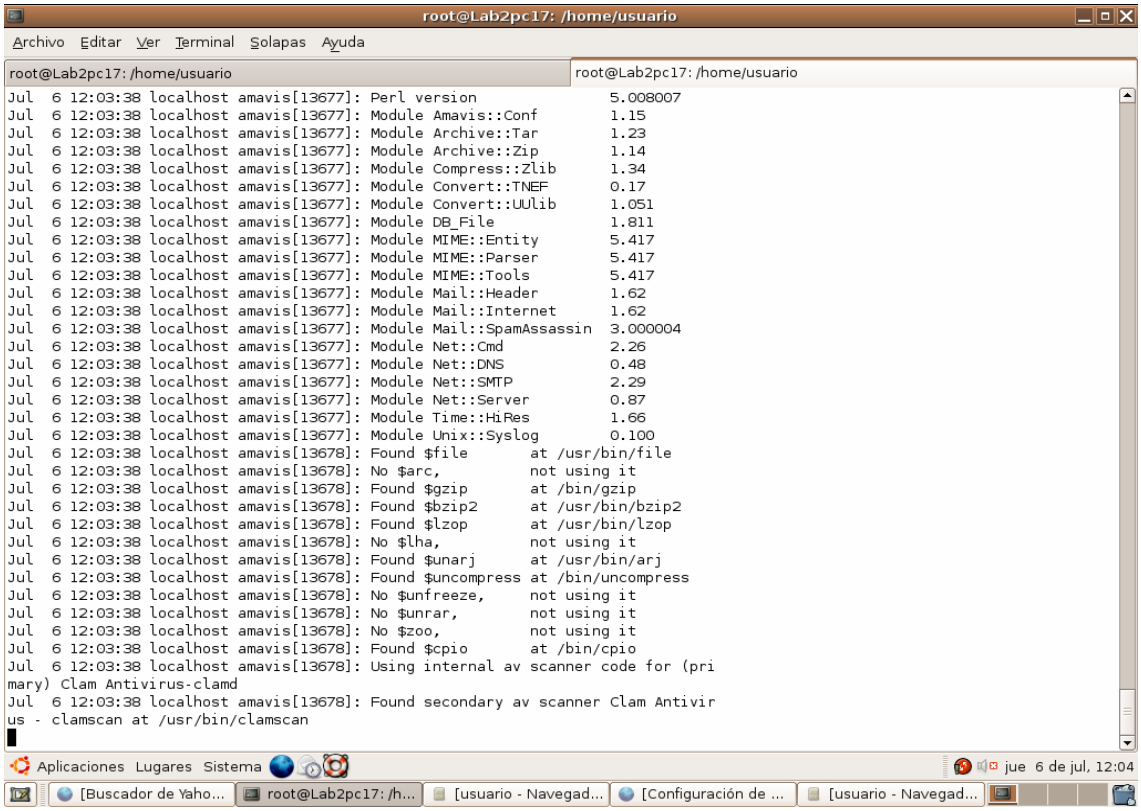
- **sa_spam_subject_tag:** Al comentar esta sentencia se desactiva la modificación del asunto del mensaje, pues con las cabeceras que se han añadido es suficiente para que *Sieve* (o nuestro cliente de correo) filtre adecuadamente.

- **banned_filename_re:** Rechazamos correos que contengan ficheros adjuntos con alguna de las extensiones mencionadas en esta variable (únicamente se permiten ficheros comprimidos), principalmente ejecutables y scripts.

Tras esto ya podemos reiniciar el servicio mediante el comando:

/etc/init.d/amavis restart

Observar su carga en el log ***/var/log/mail.log*****,** donde se informa de todos los módulos cargados al iniciar.



Hay cuatro tipos de restricciones principales, y a cada tipo de restricción se le pueden agregar varias bases de datos de comprobación.

- **smtpd_client_restrictions:** Una lista de clientes desde los que no se quiere recibir ningún correo.
- **smtpd_helo_restrictions:** Una lista de hosts desde los cuales no se quiere aceptar información helo.
- **smtpd_sender_restrictions:** Una lista de remitentes (direcciones de correo o dominios) desde los que no se quiere recibir ningún correo.
- **smtpd_recipient_restrictions:** Una lista de destinatarios (direcciones de correo o dominios) desde los que no se quiere recibir ningún correo. Esta lista de restricciones se aplica cuando el cliente envía el comando RCPT TO: Es entonces cuando ya se sabe la IP del cliente, la dirección del remitente (MAIL FROM:), y el dominio HELO.

Modificaciones en Postfix

Las modificaciones a realizar son las siguientes en el fichero **/etc/postfix/master.cf** añadimos las siguientes líneas:

```
127.0.0.1:10025 inet  n       -       n       -       -       smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_client_restrictions=
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
        -o strict_rfc821_envelopes=yes
        -o smtpd_error_sleep_time=0
```

```
    -o smtpd_soft_error_limit=1001
    -o smtpd_hard_error_limit=1000
smtp-amavis unix - - n - 2 lmtp
    -o lmtp_data_done_timeout=1200
    -o lmtp_send_xforward_command=yes
```

Con esto escucharemos por el puerto 10025 de forma local (no aceptaremos conexiones desde otras maquinas), dado que por ahí será por donde recibiremos los mails ya analizados de **amavis-new.**

Es necesario hacer algunas modificaciones en el fichero **/etc/postfix/main.cf**

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Esto se añadirá al final del archivo "/etc/postfix/main.cf" la línea "content_filter = smtp-amavis: [127.0.0.1]:10024″. Así se enviará todo mail recibido a amavis, el cual escucha por el puerto 10024 en nuestra máquina.

Una vez ***Amavisd-new*** haya finalizado su trabajo, devolverá el mensaje a Postfix a través del puerto 10025, donde hemos habilitado un *smtpd*.

Reiniciamos los servicios y todo estará listo:

 # /etc/init.d/amavis restart
 # /etc/init.d/postfix restart

8. Medidas anti-UCE

Las medidas **anti-UCE** (del inglés, *Unsolicited Commercial Email*) son una serie de mecanismos que se habilitarán en Postfix para filtrar el uso abusivo de nuestro servidor y tratar de denegar la entrada de una buena parte del correo no solicitado en él. En primer lugar, añadimos las siguientes líneas al fichero ***/etc/postfix/main.cf*:**

```
smtpd_helo_required = yes
disable_vrfy_command = yes

 smtpd_recipient_restrictions =
    reject_invalid_hostname,
    reject_non_fqdn_hostname,
    reject_non_fqdn_sender,
    reject_non_fqdn_recipient,
    reject_unknown_sender_domain,
    reject_unknown_recipient_domain,
    permit_mynetworks,
    permit_sasl_authenticated,
    reject_unauth_destination,
    check_recipient_access pcre:/etc/postfix/recipient_checks.pcre,
    check_helo_access hash:/etc/postfix/helo_checks,
    # check_helo_access pcre:/etc/postfix/helo_checks.pcre,
    check_sender_access hash:/etc/postfix/sender_checks,
    check_client_access hash:/etc/postfix/client_checks,
    check_client_access pcre:/etc/postfix/client_checks.pcre,
    reject_rbl_client relays.ordb.org,


    # reject_rbl_client opm.blitzed.org,
    # reject_rbl_client list.dsbl.org,
    # reject_rbl_client sbl.spamhaus.org,
    # reject_rbl_client cbl.abuseat.org,
    # reject_rbl_client dul.dnsbl.sorbs.net,
    permit

smtpd_data_restrictions =

    reject_unauth_pipelining,
    permit
```

De este modo, a falta de especificar los ficheros referenciados, establecemos el siguiente flujo en las restricciones aplicadas al smtp_recipient_restrictions (nótese que el orden es relevante):

1. Las primeras sentencias nos aseguran que el proceso de *HELO/EHLO* y el envoltorio del mensaje son correctos. Y en la última deshabilitamos las verificaciones de direcciones de correo.

2. Deshabilitamos la concatenación (del inglés, *pipelining*) de comandos (generalmente sólo los *spammers* tratan de concatenarlos, particularmente durante ataques de diccionario).
3. Permitimos cualquier cosa que pase las restricciones de más arriba y que pertenezca a *mynetworks* (el destino no importa).
4. Permitimos a los clientes que se han autenticado por **SASL**.
5. Rechazamos clientes sin autenticar.
6. Comprobamos ciertas direcciones de destinatarios antes de aplicar cualquier lista negra local o de DNS.
7. Comprobamos las listas negras locales, las listas blancas locales y las listas negras y blancas combinadas (comprobación de HELO/EHLO, remitente (origen en el envoltorio) y cliente (servidor que envía)).
8. Comprobamos las listas negras de DNS y de hosts que permiten relay abiertamente.

De los seis servidores de listas negras de DNS y RHS cinco aparecen comentados. Esto es debido a que, particularmente, me es suficiente el primero, que además lista únicamente servidores que hacen relay abierto (técnicamente es una respuesta de sí o no con criterios totalmente objetivos). Queda a merced del lector aplicar más o menos listas negras, pero, en cualquier caso, es muy recomendable pasarse primero por sus respectivas webs y tener bien claro qué criterios siguen para considerar a un servidor de correo como digno de aparecer en sus listas negras.

Mediante la directiva *check_recipient_access* aplicamos sobre el destinatario del mensaje las comprobaciones detalladas en el fichero ***/etc/postfix/recipient_checks.pcre***, que se resumen en revisar la sintaxis de las direcciones:

```
# This file requires PCRE support built into Postfix
/^\@/           550 Invalid address format.
/[!%\@].*\@/    550 This server disallows weird address syntax.
/^postmaster\@/ OK
/^hostmaster\@/ OK
/^abuse\@/      OK
```

Nótese que para poder referenciar ficheros con expresiones regulares, por ejemplo en él ***/etc/postfix/main.cf*** la línea ***pcre:/etc/postfix/recipient_checks.pcre***, es necesario tener instalado el paquete ***postfix-pcre***. Por otra parte, las referencias del tipo ***dbm:/etc/postfix/helo_checks*** requieren de la ejecución del comando *postmap <fichero>* para que se cree el fichero en formato Berkeley Database con extensión.*db*.

En la directiva *check_helo_access* se lleva a cabo una comprobación muy sencilla pero muy eficiente a la hora de evitar el *spam*. El contenido de ***/etc/postfix/helo_checks*** es el que sigue:

```
#Comprobaciones de HELO/EHLO

marana.isi.unanleon.edu.ni              REJECT You are not in dominio.com :C

webmail.marana.isi.unanleon.edu.ni REJECT You are not mail.dominio.com

localhost                                       REJECT You are not me :C
```

Aunque parezcan algo triviales, estas comprobaciones son enormemente efectivas y sencillas. Simplemente se verifica que el comando *HELO/EHLO* enviado por el host que se conecta a nuestro servidor no use ni nuestro dominio, ni nuestra IP pública o privada, ni *localhost*. Otra comprobación que podría hacerse sobre el comando *HELO/EHLO* sería la de la sintaxis del host, que debe cumplir el estándar del RFC. Esta comprobación aparece comentada en el ***main.cf*** de más arriba, pues no la uso en la actualidad. En cualquier caso, éste sería el contenido del fichero referenciado:

```
# This file requires PCRE support built into Postfix
/^[0-9]+(\.[0-9]+){3}$/       REJECT Invalid hostname
```

Con la directiva *check_sender_access* podemos realizar comprobaciones sobre el remitente del mensaje. En mi caso particular, este fichero únicamente contiene el comentario inicial, pues no le doy uso alguno, pero puede que el lector quiera darle algún tipo de uso parecido al que se propone a continuación para ***/etc/postfix/sender_checks*:**

```
# A compilar con postmap ...

spammers.com 554 Spam not tolerated here

someuser@morespammers.comsomeuser@morespammers.com OK

morespammers.com REJECT
```

Según este contenido, rechazaríamos todo el correo de los dominios *spammers.com* y *morespammers.com* excepto el del usuario *someuser@morespammers.com*.

Mediante la directiva *check_client_access* podemos realizar comprobaciones sobre el cliente que envía el mensaje. En el *main.cf* propuesto más arriba aparecen dos referencias a esta directiva, una usando una tabla de hashing y otras expresiones regulares. Los ficheros no contienen más que la línea de comentario inicial pero, de todas maneras, a continuación se proponen posibles usos. Este sería un ejemplo de contenido para el fichero ***/etc/postfix/client_checks*:**

```
# A compilar con postmap ...

spammers.com          554 Spam not tolerated here

10                    554 Go away!

myfriendsdomain.com OK

172.16                OK
```

De este modo, pese a que el dominio *myfriendsdomain.com* y la IP *172.11.0.0/16* pertenezcan a listas negras, nosotros decidimos aceptar los accesos que venga de ellos. Asimismo, rechazamos todas las conexiones que provengan del dominio *spammers.com* y del rango de IPs *10.0.0.0/8*. Con expresiones regulares podemos conseguir resultados más complejos, como los del fichero ***/etc/postfix/client_checks.pcre*:**

```
# This file requires PCRE support built into Postfix
/10\.9\.8\.7/           OK
/10\.9\.([89]|10)\.\d+/  554 Go away. We don't want any!
```

Esta expresión regular rechazaría las conexiones desde el rango 10.9.8.0 - 10.9.10.255 excepto la dirección 10.9.8.7.

Tras ejecutar los comandos ***postmap*** necesarios sobre los ficheros *hash* tan sólo nos queda reiniciar Postfix con el comando ***/etc/init.d/postfix restart***.

Algunas notas acerca de las restricciones y las listas de acceso sobre "hostname", "helo", "client", "sender" y "recipient"

HELO/EHLO es lo que la máquina que envía le dice que es a nuestra máquina. Puede disfrazarse fácilmente y frecuentemente se configura de manera incorrecta. *HELO/EHLO* se comprueba a través de las restricciones en el *helo* y el *hostname* del *smtpd*.

Sender es el envoltorio de la dirección remitente (*Mail From* en la comunicación ***SMTP***), no la dirección IP de la máquina cliente ni su *hostname*, ni tampoco el campo *From:* de las cabeceras (aunque el envoltorio del remitente perfectamente puede ser el mismo que el *From:* de las cabeceras). *Sender* se comprueba mediante las restricciones del remitente en el *smtpd*.

Client es la dirección IP de la máquina que realiza el envío, y posiblemente el *hostname* (sí puede obtenerse alguno de la resolución inversa de la dirección IP). *Client* se comprueba con las restricciones del cliente en el *smtpd*.

Recipient hace referencia a la dirección de correo pasada en el comando *RCPT TO* durante la comunicación ***SMTP***, no el *To:* ni el *Ccc:* u otros campos de las cabeceras. *Recipient* se comprueba mediante las restricciones del destinatario en el ***smtpd***.

Si se sitúan las listas de acceso antes de las comprobaciones de listas negras de DNS, tal y como se muestra en la configuración del *main.cf* más arriba, pueden servir tanto como listas negras que como listas blancas. Pero es muy importante ser cauteloso a la hora de usar las listas blancas pues, por ejemplo, si se le da él *OK* a algo en la directiva *smtpd_recipient_restrictions* antes del *reject_unauth_destination*, podríamos dejar el servidor haciendo relay abierto para todo aquel que tenga él *OK*, o a los que sean falsificables.

9. Configuración de apache2 con SSL

Instalamos Apache2:

apt-get install apache2

Habilitamos el módulo ssl:

a2enmod ssl

Ejecutamos un script para crear nuestro certificado de seguridad para el servidor (estará autofirmado).

 # apache2-ssl-certificate

Nos hará una serie de preguntas…

```
# apache2-ssl-certificate

creating selfsigned certificate
replace it with one signed by a certification authority (CA)

enter your ServerName at the Common Name prompt

If you want your certificate to expire after x days call this programm
with -days x
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to '/etc/apache2/ssl/apache.pem'
-----
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) [GB]:NI
State or Province Name (full name) [Some-State]:Leon
Locality Name (eg, city) []:Leon
Organization Name (eg, company; recommended) []:Unan-leon
Organizational Unit Name (eg, section) []:
server name (eg. ssl.domain.tld; required!!!) []:
Email Address []:
```

Ahora crearemos la configuración del sitio para el servidor seguro basándonos en la que lleva por defecto:

 # cp /etc/apache2/sites-available/default /etc/apache2/sites-available/ssl
 # ln -s /etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl

/etc/apache2/sites-enabled/ssl tiene que empezar de la siguiente manera:

```
NameVirtualHost *:443

<VirtualHost *:443>
   ServerAdmin webmaster@localhost

   DocumentRoot /usr/share/squirrelmail
   <Directory />
      Options FollowSymLinks
      AllowOverride None
   </Directory>
   <Directory /var/www/ssl.webmail.marana.isi.unanleon.edu.ni>
#[...aquí sigue...]
```

Ahora, **/etc/apache2/sites-enabled/default** también hay que configurarlo de la misma forma:

```
NameVirtualHost *:80
<VirtualHost *:80>
   ServerAdmin webmaster@localhost


   DocumentRoot /usr/share/squirrelmail
   <Directory />
      Options FollowSymLinks
      AllowOverride None
   </Directory>
   <Directory /var/www/webmail.marana,isi.unanleon.edu.ni/htdocs>
#[...aquí sigue...]
```

Ahora añade en el fichero /**etc/apache2/ports.conf**:

```
Listen 443
```

Por último, sólo basta añadir dentro del fichero “/etc/apache2/sites-enabled/ssl” en cualquier lugar (por ejemplo justo debajo de “ServerSignature On”):

```
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/apache.pem
```

Y por último, reiniciamos apache2:

 # /etc/init.d/apache2 force-reload

10. Listas de correo con Mailman

Mailman es un programa que le permite administrar listas de correo electrónico, con soporte para un rango amplio de tipos de listas de correo, tales como listas de discusión general y listas de sólo anuncios.
Su interfaz integrada ofrece un acceso fácil a la lista de miembros y de administradores.

Soporta almacén de archivos integrado, proceso automático de rechazos, filtro de contenido, envío del compendio de artículos publicados, filtros contra el spam, etc.

Características especiales:

- Permite la creación y eliminación de las listas a través de la web (con soporte automático dependiendo del **MTA**).
- Soporte multi idioma, configurable por sitio, por lista y por usuario
- Soporta "Real name" para los miembros.
- Soporta envíos personalizados.
- Moderación de emergencia.
- Contenido filtrado basado en **MIME**.
- Administración de membresías y búsqueda incluida.
- Soporte para grupos de noticias moderados.
- Invitaciones.
- Autorespuestas.

Es conveniente reflexionar sobre si se desea acceder a su gestión Web a través de *http* o de *https* (será necesaria la instalación del soporte **SSL** para Apache) y si se desea instalar la versión 1 o 2 de Apache. Antes de ejecutar el comando también conviene asegurarse de que tenemos un alias para *root* en el */etc/aliases* apuntando a un usuario que tenga buzón de correo (p.e. martha).

#pico *etc/aliases*

Primero procederemos a instalar el paquete de **Mailman** mediante el comando:

 # apt-get install mailman

En el diálogo de configuración que nos presenta el script de postinstalación elegimos los idiomas que vamos a utilizar en **Mailman** y, de entre ellos, el que será elegido por defecto. A continuación debemos modificar ligeramente el fichero ***/etc/mailman/mm_cfg.py*** a fin de que **Mailman** sepa que se está trabajando con **Postfix** como *Mail Transport Agent*:

Descomentamos donde dice:

MTA = 'None'

Poniendo:

```
MTA = 'Postfix'
```

La configuración por defecto de **Postfix** nos deja la directiva *alias_maps* apuntando a */etc/aliases*. Ya que no nos interesa estar modificando este fichero y ejecutando el comando *newaliases* de **Postfix** cada vez que creemos o borremos una lista, utilizaremos el fichero de alias propio de **Mailman**, que es automáticamente actualizado por los comandos *newlist* y *rmlist*. El primer paso será generarlo:

 # cd /var/lib/mailman
 # bin/genaliases

A continuación añadiremos ese fichero de alias a la directiva *alias_maps* del ***/etc/postfix/main.cf***, además de otras directivas necesarias, tal que:

```
alias_maps = hash:/etc/aliases, hash:/var/lib/mailman/data/aliases

mailman_destination_recipient_limit = 1
unknown_local_recipient_reject_code = 550
owner_request_special = no
recipient_delimiter = +
```

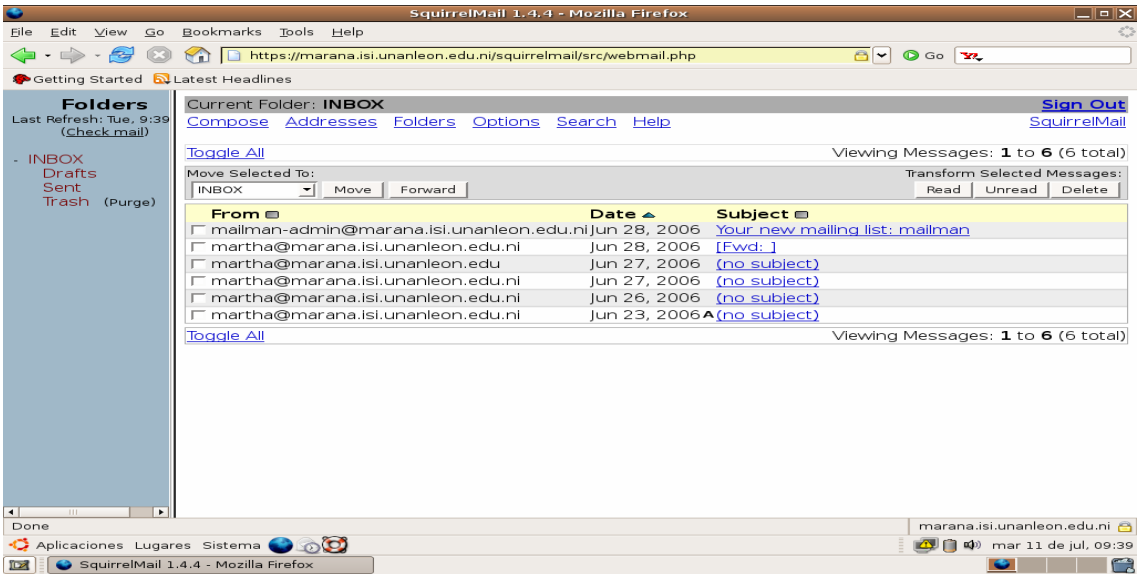
Y solicitaremos a **Postfix** que recargue la configuración:

 # /etc/init.d/postfix reload

El otro paso de la instalación de **Mailman** nos avisa que es necesario crear una *site list* llamada ***mailman*** y que hasta que no la creemos el demonio del Mailman no arrancará. Ahora es el momento de crearla y, para ello, ejecutamos el siguiente comando:

 # newlist mailman
Enter the email of the person running the list: martha@marana.isi.unanleon.edu.ni
Initial mailman password:
Hit enter to notify mailman owner...

Y pasamos a iniciar el demonio de **Mailman** mediante el comando **/etc/init.d/mailman start**. Una vez iniciado el servicio, recibiremos el correo que nos notifica la creación de la lista en la dirección de correo que hayamos especificado (martha@marana.isi.unanleon.edu.ni)



#/etc/init.d/postfix reload.

Finalmente, para poder acceder a la interfaz web de **Mailman**, deberemos llevar a cabo unas simples modificaciones en Apache:

1. **Activar el módulo *cgi*,** mediante el comando *a2enmod cgi*. .
2. **Crear un fichero llamado *mailman*** dentro del directorio */etc/apache2/conf.d/* que contenga estas tres líneas.

```
ScriptAlias /mailman/ /usr/lib/cgi-bin/mailman/
Alias /pipermail/ /var/lib/mailman/archives/public/
Alias /images/mailman/ /usr/share/images/mailman/
```

Reiniciar Apache 2 con *apache2ctl graceful*, podemos acceder a nuestra interfaz.

http://marana.isi.unanleon.edu.ni/mailman/listinfo

Conclusión

Hemos logrado instalar y configurar nuestro servidor de Correo Electrónico e implementar y probar la funcionalidad del servidor, con los mecanismos de seguridad asociados.

Cabe resaltar que nuestro proyecto se realizó bajo el subdominio de la UNAN-LEON, el cual lleva por nombre marana.isi.unanleon.edu.ni. Las instalaciones y configuraciones realizadas funcionan en este nuestro caso, para la universidad, pero funciona igual en cualquier otra institución que posea un dominio.

Es importante resaltar que los elementos fundamentales de estos servicios de correo electrónico es el Cifrado de la Información y la Autenticación de los usuarios, mediante TLS (Transport Layer Security) y SASL (Simple Autentication and Security Layer).

Recomendaciones

En base a las experiencias obtenidas en el transcurso de nuestro trabajo, se podrían implementar mejoras en el servidor para futuros proyectos como:

- Otros mecanismos de autenticación como PAM, LDAP, MD5 etc.
- Alias virtuales.
- Dominios Virtuales.
- Distribuciones como Suse, Red Hat, Debian entre otras.
- En el caso de que el servidor SMTP Postfix y el servidor IMAP Cyrus deban residir en maquinas diferentes será necesario usar sockets TCP en lugar de sockets Unix para conectarlos.
- Un visor de Logs, como por ejemplo Mailgraph que genera gráficos diarios, mensuales, semanales y anuales de los mails recibidos, enviados, rebotado, con spam y con virus.
- Se pueden usar otros servicios de Webmail como por ejemplo horde.

Glosarios de Términos

Alias: Apodo o Pseudónimo. Nombre usualmente corto y fácil de recordar que se utiliza en lugar de otro nombre usualmente largo y difícil de memorizar.

Apache: Es programa de servidor HTTP Web de código abierto (open source). Fue desarrollado en 1995 y actualmente es uno de los servidores Web más utilizados en la red. Usualmente corre en UNIX, Linux, BSD y Windows. Es un poderoso paquete de servidor Web con muchos módulos que se le pueden agregar y que se consiguen gratuitamente en el Internet. Uno de sus competidores es Microsoft IIS. http://www.apache.org.

@ **(arroba):** Signo que forma parte de las direcciones de correo electrónico de forma que separa el nombre del usuario de los nombres de dominio del servidor de correo (ejemplo info @ panamacom.com). Su uso en Internet se origina en su frecuente empleo como abreviatura de la preposición Inglesa at (en).

Backup: Copia de seguridad. Acción de copiar documentos, archivos o ficheros de tal forma que puedan recuperarse en caso de fallo en el sistema.

Cliente: Aplicación que permite a un usuario obtener un servicio de un servidor localizado en la red. Sistema o proceso el cual le solicita a otro sistema o proceso la prestación de un servicio.

CAcert: Es una organización certificadora sin ánimo de lucro que pretende promocionar el conocimiento y la educación en la seguridad informática a través del uso de cifrado, especialmente la familia X.509 de estándares. Esta organización puede firmar nuestro certificado digital, pero aún no es una autoridad certificadora reconocida, por lo cual será siempre necesario instalar su certificado raíz en la aplicación cliente. En cualquier caso, en la actualidad no hay diferencia entre usar un certificado firmado por esta asociación o por una autoridad certificadora creada por nosotros mismos, pero quizás en el futuro esto cambie.

Daemon: Aplicación UNIX la cual está permanentemente en estado de alerta en un servidor Internet con el fin de realizar determinadas tareas como, por ejemplo, enviar un mensaje de correo electrónico o servir una página Web.

Drivers: Existen muchos periféricos que se pueden conectar a un ordenador (disqueteras, impresoras, lectores de CD, escaners, etc). Para que el sistema sea capaz de aprovechar al máximo las capacidades de cada uno de estos dispositivos, los fabricantes incluyen unos programas llamados "Drivers", que son los que saben gestionar adecuadamente ese periférico.

DNS: Servidor de Nombres de Dominio. Servidor automatizado utilizado en el Internet cuya tarea es convertir nombres fáciles de entender (como www.panamacom.com) a direcciones numéricas de IP.

Dominio: Sistema de denominación de hosts en Internet el cual está formado por un conjunto de caracteres el cual identifica un sitio de la red accesible por un usuario. Los dominios van separados por un punto y jerárquicamente están organizados de derecha a izquierda. Comprenden una red de computadoras que comparten una característica común, como el estar en el mismo país, en la misma organización o en el mismo departamento. Cada dominio es administrado por un servidor de dominios. Los dominios se establecen de acuerdo al uso que se le da a la computadora y al lugar donde se encuentre. Los más comunes son .com, .edu, .net, .org y .gov; la mayoría de los países tienen su propio dominio, y en la actualidad se están ofreciendo muchos dominios nuevos debido a la saturación de los dominios .com (utilizados muchas por empresas).

e-mail: El e-mail, de las palabras inglesas electronic mail (correo electrónico), es uno de los medios de comunicación de más rápido crecimiento en la historia de la humanidad y más usados en Internet. Por medio del protocolo de comunicación TCP/IP, permite el intercambio de mensajes entre las personas conectadas a la red de manera similar al correo tradicional. Para ello es necesario disponer de una dirección de correo electrónico, compuesta por el nombre del usuario, la arroba "@" y el nombre del servidor de correo. Por ejemplo, sample@panamacom.com, donde 'sample' es el usuario y panamacom.com el nombre del host o servidor.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

FTP: File Transfer Protocol. Protocolo de transferencia de archivos. Se usan programas servidores de FTP como por ej. NcFTPd. Estos programas permiten la conección entre dos computadoras, usando por lo general el puerto 21 para conectarse (aunque se puede usar otros puertos). Por medio del Protocolo de transferencia de archivos se pueden uploadear y downloadear archivos entre el cliente y el host (servidor).

Gateway: El significado técnico se refiere a un hardware o software que traduce dos protocolos distintos o no compatibles. Gateway o pasarela es un dispositivo, con frecuencia un ordenador, que realiza la conversión de protocolos entre diferentes tipos de redes o aplicaciones. Por ejemplo, un

gateway de correo electrónico, o de mensajes, convierte mensajes entre dos diferentes protocolos de mensajes.

GPL: General Public License -- Licencia de regulación de los derechos de autor de los programas de software libre (free software) la cual es promovida por la Free Software Foundation (FSF) en el marco de la iniciativa GNU. Permite la distribución de copias de programas (e incluso cobrar por ello), así como modificar el código fuente de los mismos o utilizarlo en otros programas.

Gusanos: Estos virus no se copian dentro del código de otros ficheros sino que se copian ellos mismos. Los gusanos más frecuentes son los que se copian utilizando la libreta de direcciones de Microsoft Outlook. Se envían a sí mismos como ficheros adjuntos.

Host: Servidor que nos provee de la información que requerimos para realizar algún procedimiento desde una aplicación cliente a la que tenemos acceso de diversas formas (ssh, FTP, www, email, etc.). Al igual que cualquier computadora conectada a Internet, debe tener una dirección o número IP y un nombre.

HTTP: En inglés Hypertext Transfer Protocol. Protocolo de Transferencia de Hipertexto. HTTP es un protocolo con la ligereza y velocidad necesaria para distribuir y manejar sistemas de información hipermedia. HTTP ha sido usado por los servidores World Wide Web desde su inicio en 1993.

HTTPS: Creado por Netscape Communications Corporation para designar documentos que llegan desde un servidor Web seguro. Esta seguridad es dada por el protocolo SSL (Secure Socket Layer) basado en la tecnología de encriptación y autenticación desarrollada por RSA Data Security Inc.

Internet: Una red mundial, de redes de computadoras. Es una interconexión de redes grandes y chicas alrededor del mundo. El Internet empezó en 1962 como una red para los militares llamada ARPANet, para que en sus comunicaciones no existan “puntos de falla”. Con el tiempo fue creciendo hasta convertirse en lo que es hoy en día, una herramienta de comunicación con decenas de miles de redes de computadoras unidas por el protocolo TCP/IP. Sobre esta red se pueden utilizar múltiples servicios como por ejemplo emails, WWW, etc. que usen TCP/IP.

IP: Internet Protocol, Protocolo de Internet. Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet. El IP es la dirección numérica de una computadora en Internet de forma que cada dirección electrónica se asigna a una computadora conectada a Internet y por lo tanto es única. La dirección IP esta compuesta de cuatro octetos como por ejemplo, 132.248.53.10

IMAP IDLE: Es una extensión de IMAP que sirve para que el servidor avise al cliente cuando ha llegado un correo y se sincronicen. La alternativa sería que el cliente lea cada poco tiempo el servidor para ver si hay correos.

IMAP: Protocolo de Acceso a Mensajes de Internet. IMAP es un acrónimo inglés de Internet Message Access Protocol. Diseñado con el fin de permitir la manipulación de buzones remotos como si fueran locales. IMAP requiere de un servidor que haga las funciones de oficina de correos pero en lugar de leer todo el buzón y borrarlo, solicita sólo los encabezados de cada mensaje. Se pueden marcar mensajes como borrados sin suprimirlos completamente, pues estos permanecen en el buzón hasta que el usuario confirma su eliminación. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet. Una vez configurada la cuenta IMAP, puede especificar las carpetas que desea mostrar y las que desean ocultar, esta característica lo hace diferente del protocolo POP.

Kernel: Núcleo. Parte fundamental de un programa, por lo general de un sistema operativo, que reside en memoria todo el tiempo y que provee los servicios básicos. Es la parte del sistema operativo que está más cerca de la máquina y puede activar el hardware directamente o unirse a otra capa de software que maneja el hardware.

LAN: Local Área Network. Red de área local. Red de computadoras personales ubicadas dentro de un área geográfica limitada que se compone de servidores, estaciones de trabajo, sistemas operativos de redes y un enlace encargado de distribuir las comunicaciones. Por ejemplo, computadoras conectadas en una oficina, en un edificio o en varios. Se pueden optimizarse los protocolos de señal de la red hasta alcanzar velocidades de transmisión de 100 Mbps.

Lista de Correo: Mailing List. Listado de direcciones electrónicas utilizado para distribuir mensajes a un grupo de personas y generalmente se utiliza para discutir acerca de un determinado tema. Una lista de distribución puede ser abierta o cerrada y puede tener o no un moderador. Si es abierta significa que cualquiera puede suscribirse a ella; si tiene un moderador los mensajes enviados a la lista por cualquier suscriptor pasan primero por aquel, quien decidirá si distribuirlos o no a los demás suscriptores.

Login: Clave de acceso que se le asigna a un usuario con el propósito de que pueda utilizar los recursos de una computadora. El login define al usuario y lo identifica dentro de Internet junto con la dirección electrónica de la computadora que utiliza.

Mail: Programa en ambiente UNIX para la edición lectura y respuesta de emails.

MTA (Mail Tranfer Agent): Es el programa encargado de recoger mensajes y enviarlos, y de comunicarse con otros MTA si es necesario.

MUA (Mail User Agent): Es un programa cliente que solicita la descarga de correos de un servidor y él envió hacia él mismo para su futura entrega a otros servidores.

MDA (Mail Delivery Agent): Es el programa encargado de recibir los correos del MTA para la entrega local y colocarlos adecuadamente en los buzones individuales de cada usuario (si el usuario tiene una cuenta en el servidor local).

MUD: Multi-user Dimensión. Dimensión Multi-Usuario. Entorno de realidad virtual, basado en texto o gráficos, en el cual los usuarios pueden conversar o interpretar diferentes roles como diversión. Los usuarios entran en el juego desde cualquier parte de Internet y solo tienen que conectarse por medio de la red al sistema donde se guarda el juego para posteriormente interactuar de manera recíproca uno con otro.

MIME: (Multipurpose Internet Mail Extensions, Extensiones de Correo Internet Multipropósito), son una serie de convenciones o especificaciones dirigidas a que se puedan intercambiar a través de Internet todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos

Network: Sistema de comunicación de datos que conecta entre sí sistemas informáticos situados en lugares más o menos próximos. Puede estar compuesta por diferentes combinaciones de diversos tipos de redes.

Networking: Término utilizado para referirse a las redes de telecomunicaciones en general.

OSI: Interconexión de Sistemas Abiertos (Open Systems Interconnect). Es el protocolo en el que se apoya Internet. Establece la manera como se realiza la comunicación entre dos computadoras a través de siete capas: Física, Datos, Red, Transporte, Sesión, Presentación y Aplicación.

Paquete: Un paquete es un pedazo de información enviada a través de la red. La unidad de datos que se envía a través de una red la cual se compone de un conjunto de bits que viajan juntos. En Internet la información transmitida es dividida en paquetes que se reagrupan para ser recibidos en su destino. Ver también conmutación de paquetes.

Password: Código utilizado para accesar un sistema restringido. Pueden contener caracteres alfanuméricos e incluso algunos otros símbolos. Se destaca que la contraseña no es visible en la pantalla al momento de ser tecleada con el propósito de que sólo pueda ser conocida por el usuario.

Pentium: Microprocesador de 64 bits, sucesor del chip 80468, de la empresa Intel. Lo llamaron así puesto que la corte Norteamericana no aceptó 586 o 80586 como marca registrada. Fue lanzado al mercado en 1993. Al pasar los años, Pentium ha evolucionado a P1, P2, P3 y P4, P4EE.

PERL: Practical Extraction and Report Language -- Lenguaje de programación muy utilizado para la elaboración de aplicaciones CGI, principalmente para realizar consultas a bases de datos como Oracle, SQL-Server, SyBase, etc, o a herramientas locales como WAIS. Perl es un lenguaje para manipular textos, archivos y procesos, proporciona una forma fácil y legible para realizar trabajos que normalmente se realizarían en C o en un shell. Perl nació y se ha difundido bajo el sistema operativo UNIX, aunque existe para otras plataformas. Perl fue desarrollado por Larry Wall, y está distribuido libremente bajo la filosofía de la GNU.

Protocolo: Descripción formal de formatos de mensaje y de reglas que dos computadoras deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina a máquina o intercambios de alto nivel entre programas de asignación de recursos.

POP: Post Office Protocol (Protocolo de Oficina de Correos). Programa cliente que se comunica con el servidor de forma que identifica la presencia de nuevos mensajes, solicita la entre de los mismos y utiliza al servidor como oficina despachadora de correo electrónico cuando el usuario envía una carta. Los mensajes enviados a la aplicación cliente son inmediatamente eliminados del servidor, sin embargo, pueden omitir este paso. La última versión, POP3, es la más utilizada.

Puerto: Son los puntos de enganche para cada conexión de red que realizamos. El protocolo TCP (el utilizado en internet) identifica los extremos de una conexión por las direcciones IP de los dos nodos (ordenadores) implicados (servidor y cliente) y el número de los puertos de cada nodo.

Postmaster: Administrador de Correos. Persona responsable de solucionar problemas en el correo electrónico, responder a preguntas sobre usuarios así como otros asuntos de una determinada instalación.

Proxy: Es un programa que realiza la tarea de encaminador, utilizado en redes locales, su función es similar a la de un router, pero es injustificable el gasto en redes locales.

PHP: Es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. El nombre es el acrónimo recursivo de "PHP: Hypertext Preprocessor" y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

Plugins: Programas que se agregan a un navegador del WWW los cuales realizan funciones determinadas. Producen la visualización de archivos multimedia y dan soporte a archivos gráficos no estándares con el visualizador.

plug-in: Es un programa de ordenador que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica.

Queue: Es "una fila" de paquetes en espera de ser procesados.

Raíz (Root): Directorio inicial de un sistema de archivos mientras que en entornos UNIX también se refiere al usuario principal.

Repositorio: Es un término utilizado en el dominio de las herramientas CASE. El repositorio podría definirse como la base de datos fundamental para el diseño; no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos software necesarios para el trabajo de programación.

RFC: En inglés es Requests for Comments. Serie de documentos iniciada en 1967 la cual describe el conjunto de protocolos de Internet y experimentos similares. No todos los RFC (en realidad muy pocos de ellos) describen estándares de Internet pero todos los estándares Internet están escritos en formato RFC. La serie de documentos RFC es inusual en cuanto los protocolos que describen son elaborados por la comunidad Internet que desarrolla e investiga, en contraste con los protocolos revisados y estandarizados formalmente que son promovidos por organizaciones como CCITT y ANSI. El RFC 822 es el formato estándar Internet para cabeceras de mensajes de correo electrónico. El nombre viene del "RFC 822", que contiene esa especificación (STD 11, RFC 822). El formato 822 era conocido antes como formato 733.

Router: Un dispositivo que conecta dos redes; opera como un bridge pero también puede seleccionar rutas a través de una red.

Sendmail: Programa servidor de emails utilizado comúnmente en UNIX, FreeBSD y Linux, entre otros.

Servidor: Un servidor es una computadora que maneja peticiones de data, email, servicios de redes y transferencia de archivos de otras computadoras (clientes). También puede referirse a un software específico, como lo es el

servidor WWW. Una computadora puede tener distintos software de servidor, proporcionando muchos servidores a clientes en la red. Por ejemplo, las computadoras que contienen sitios Web se llaman servidores ya que "sirven" recursos de Web para aplicaciones cliente como los navegadores o browsers.

Servidor de Correo: Un servidor de correo (mail server) es la computadora donde se ejecuta un programa de gestión de emails, como por ejemplo Sendmail, Qmail y Microsoft Exchange.

Servidor Web: Un servidor Web es el programa, y la computadora que lo corre, que maneja los dominios y páginas Web, interpretando lenguajes como html y php, entre otros.

Socket: El protocolo TCP/IP gestiona el envío y la recepción de la información los denominados paquetes, que resultan de su división en trozos más pequeños. La biblioteca que controla el envío/recepción de estos paquetes se denomina socket.

spam: Son mensajes de correo enviados por robots (programas automáticos) que en el pasado usaban las vulnerabilidades de los MTA para ser enviados por servidores de correo. Generalmente envían publicidad, cadenas, y otras clases de basura. Los mensajes son inofensivos y sólo son fuente de distracción para los usuarios y almacenamiento inútil de espacio en disco.

Spammer: Usuario de correo que hace spam.

SSL: Acrónimo de Secure Sockets Layer. El protocolo SSL permite un intercambio de información entre el servidor Web y el navegador del usuario (cliente) de forma segura. El objetivo de esta tecnología es permitir que sólo el usuario autorizado pueda efectuar las operaciones. Mientras el protocolo HTTP utiliza un formato de dirección que empieza por http://, las direcciones en el protocolo SSL empiezan por https://.

SASL (Simple Authentication and Security Layer): Es un método para proveer a un protocolo de comunicación el soporte de autentificación. Con el empleo de SASL, un protocolo usa un comando para identificar y autentificar la conexión del usuario ante el servidor y negocia la protección a la subsecuente interacción del protocolo. SASL puede proveer autentificación a las conexiones SMTP, IMAP, POP, LDAP, LPR, SSH, etc.

SMTP (Simple Mail Transfer Protocol): Es el protocolo estándar de transmisión de información para el envío de e-mail entre servidores de correo. Adicionalmente, los clientes usan el protocolo SMTP para el envío de correo a un servidor que cuenta con un MTA instalado. La comunicación SMTP usualmente es a través del puerto 25 de comunicaciones.

Software: Se refiere a programas en general, aplicaciones, juegos, sistemas operativos, utilitarios, antivirus, etc. Lo que se pueda ejecutar en la computadora.

Software libre: Programas desarrollados y distribuidos dándole al usuario la libertad de ejecutar, copiar, distribuir, cambiar y mejorar dicho programa (Linux es un ejemplo) mediante su código fuente. El software libre no es siempre software gratuito (equivocación bastante habitual que tiene su origen de la palabra en inglés "free" que significa tanto "libre" como "gratuito").

Swapping: Es un mecanismo o modo de interrelacionar la memoria principal (la que contiene el Programa en ejecución, los datos de proceso inmediato y los resultados intermedios) con la secundaria, de tal modo que se produce un intercambio de programas entre ambas cuyo resultado es la simulación de un sistema multitarea o la potenciación de memoria central a base de recursos de la memoria secundaria.

TCP/IP: El nombre TCP/IP proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). En español es Protocolo de Control de Transmisión y Protocolo de Internet. Forma de comunicación básica que usa el Internet, la cual hace posible que cualquier tipo de información (mensajes, gráficos o audio) viaje en forma de paquetes sin que estos se pierdan y siguiendo cualquier ruta posible.

Telnet: Servicio de Internet con el cual un usuario se puede conectar de forma remota a otra computadora, como si se hiciera desde un terminal local, usualmente por el puerto 23. Es preferible usar otros programas más actualizados como ssh2, ya que telnet tiene vulnerabilidades.

TLS (Transport Layer Security): Es una versión estandarizada por el IETF del protocolo SSL que pretende abarcar toda la capa de transporte de la pila OSI. Es un nivel (Layer) que garantiza seguridad (conexión segura) y privacidad a la interacción de un protocolo ya que provee mecanismos de encripción a la comunicación. Su antecesor es SSL (Security Socket Layer). Los protocolos en nivel TLS cambian su nomenclatura y el puerto de comunicaciones que utilizan. Así http (puerto 80) sobre TLS cambia a https (puerto 443).

Troyanos: Este tipo de virus se camufla dentro de un programa que parece inofensivo e interesante, para que el usuario lo ejecute y así llevar a cabo el fin para el que fueron programados. En ocasiones lo que pretenden es sacar al exterior información de nuestro ordenador, como podrían ser contraseñas y otros tipos de datos que pudieran ser valiosos.

UNIX: Sistema operativo especializado en capacidades de multiusuario y multitarea. Fue la base inicial de Internet. Entre la característica más importantes se encuentran: Redireccionamiento de Entrada/Salida.

URL: Acrónimo de Uniform Resource Locator. Localizador Uniforme de Recurso. Es el sistema de direcciones en Internet. El modo estándar de escribir la dirección de un sitio específico o parte de una información en el Web.

User ID: Identificación de usuario. Conjunto de caracteres alfanuméricos los cuales sirven para identificar a un usuario para su acceso a algún sistema, por ejemplo Web sites, banca electrónica, emails, etc.

Usuario: Persona que tiene una cuenta en una determinada computadora por medio de la cual puede acceder a los recursos y servicios que ofrece una red. Puede ser tanto usuario de correo electrónico como de acceso al servidor en modo terminal. Un usuario que reside en una determinada computadora tiene una dirección única de correo electrónico.

Virus: Es un programa capaz de instalarse y ejecutarse por sí mismo ("reproducirse") en un equipo Windows, y usualmente causar daños a archivos y programas. Algunos virus pueden enviarse por correo electrónico, e instalarse ("infectar y propagarse") en otros equipos remotos.

Webmail: Servicio que permite gestionar el correo electrónico desde un sitio Web el cual es de gran utilidad para personas que tienen que desplazarse con frecuencia y lo ofrecen habitualmente los proveedores de acceso a Internet. Entre los más utilizados están Hotmail, Yahoo mail y Gmail.

Webmaster: Administrador de Web - Persona responsable de la gestión y mantenimiento de un servidor Web, principalmente desde el punto de vista técnico; por lo que no debe ser confundido con un editor de Web.

Xenix: Sistema operativo de Microsoft conforme con Unix.

Los RFC (Request for Comments) de Internet

Los RFC que definen el sistema de DNS están disponibles en diversos sitios Web, como RFC Editor o Internet RFC/STD/FYI/BCP Archives. Las ideas iniciales y en desarrollo aparecen primero como borradores y son más tarde formalizadas como RFC. A continuación se listan algunos de los más relevantes que hacen referencia a los formatos y protocolos usados aquí:

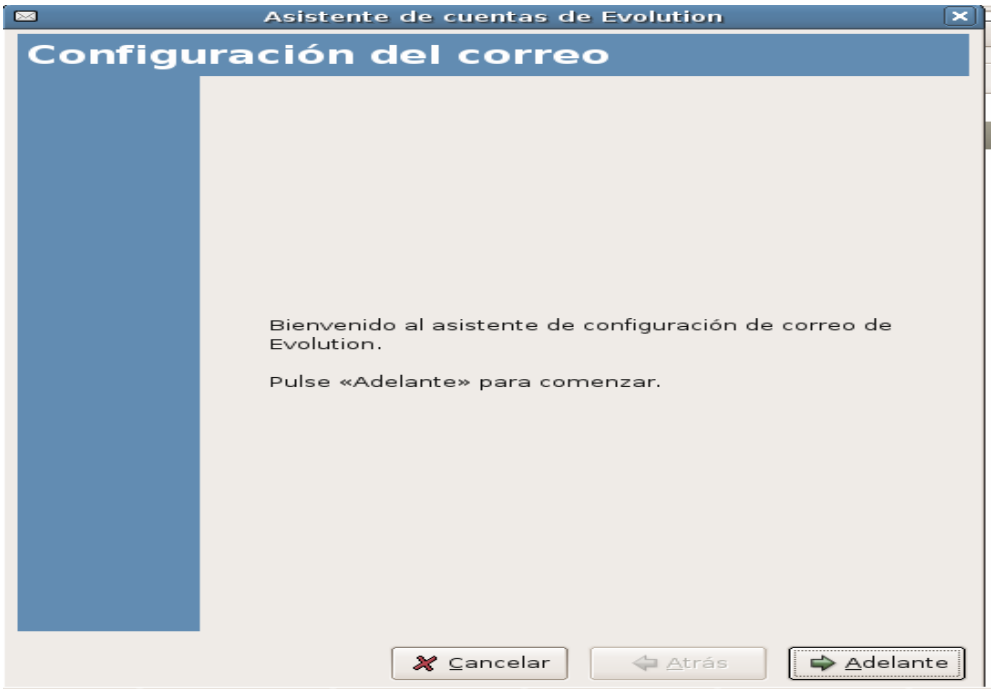
821: Simple Mail Transfer Protocol.
1651: SMTP Service Extensions.
1652: SMTP Service Extension for 8bit-MIMEtransport.
1870: SMTP Service Extension for Message Size Declaration.
1939: Post Office Protocol - Version 3.
2033: Local Mail Transfer Protocol.
2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response.
2222: Simple Authentication and Security Layer (SASL).
2245: Anonymous SASL Mechanism.
2289: A One-Time Password System.
2444: The One-Time-Password SASL Mechanism.
2487: SMTP Service Extension for Secure SMTP over TLS.
2554: SMTP Service Extension for Authentication.
2595: Using TLS with IMAP, POP3 and ACAP.
2821: Simple Mail Transfer Protocol.
2831: Using Digest Authentication as a SASL Mechanism.
2920: SMTP Service Extension for Command Pipelining.
3546: Transport Layer Security (TLS) Extensions.

Bibliografía

- http://cernicalo.escomposlinux.org/~emeteo/imap/imap+postfix.pdf
- http://cernicalo.escomposlinux.org/~emeteo/imap/imap+postfix/
- http://www.ecualug.org/?q=2005/07/13/comos/guia_de_instalacion_de_postfix_con_dovecot
- http://bulma.net/body.phtml?nIdNoticia=1621&nIdPage=2
- http://www.guia-ubuntu.org/breezy/doku.php
- http://www.guia-ubuntu.org/breezy/servidor/servidor_correo
- http://dev.mysql.com/doc/refman/5.0/en/index.html
- http://www.aet.tuottbus.de/personen/jaenicke/postfix_tls/docs/myownca.html
- http://www.seguridad.unam.mx/doc/?ap=tutorial&id=182
- http://72.14.203.104/search?q=cache:AL1T3HduM2AJ:cursos.gpltarragona.org/material-avancat/apunts/postfix.pdf+COMO+CREAR+CERTIFICADOS+PARA+POSTFIX+EN+UBUNTU&hl=es&gl=ni&ct=clnk&cd=10&lr=lang_es
- http://www.fentlinux.com/web/?q=node/358
- http://www.gpltarragona.org/2006/04/01/guia-de-referencia-rapida-para-postfix-amavis-new-clamav-spamassassin-procmail-y-dovecot-en-ubuntu/
- http://www.gpltarragona.org/archives/318

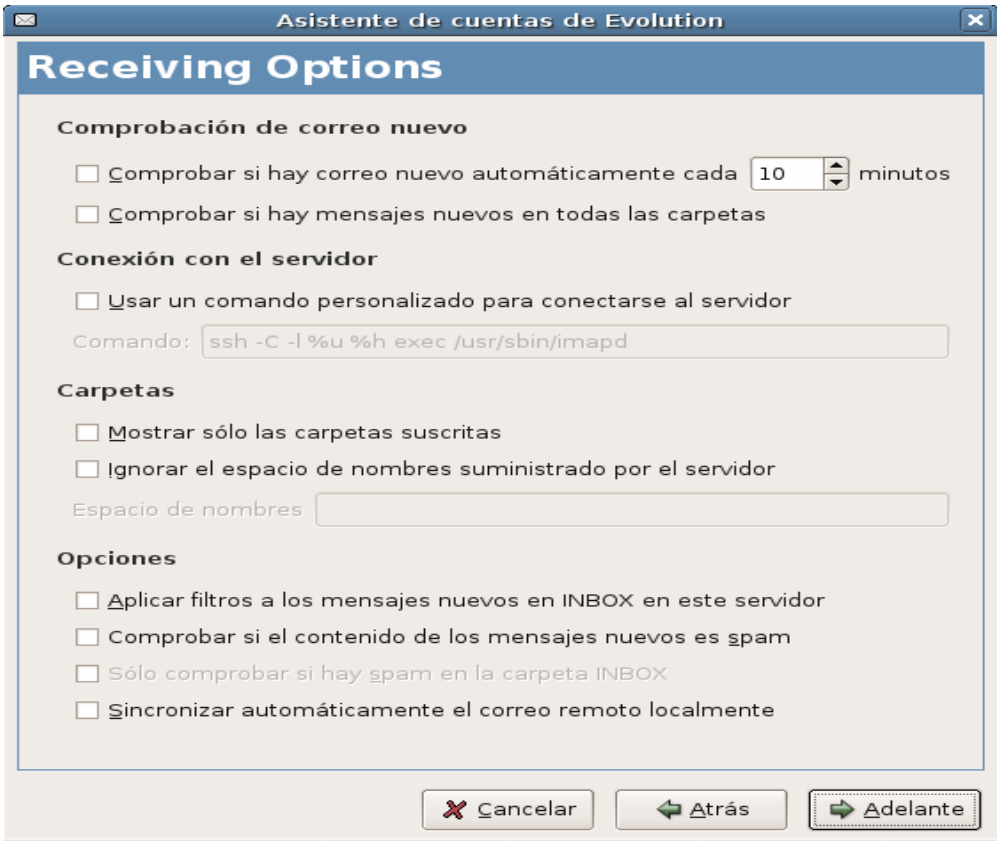
Anexos

Para revisar los correos debemos configurar un visor de correo en este caso utilizamos el evolution.



Dar clic en adelante.

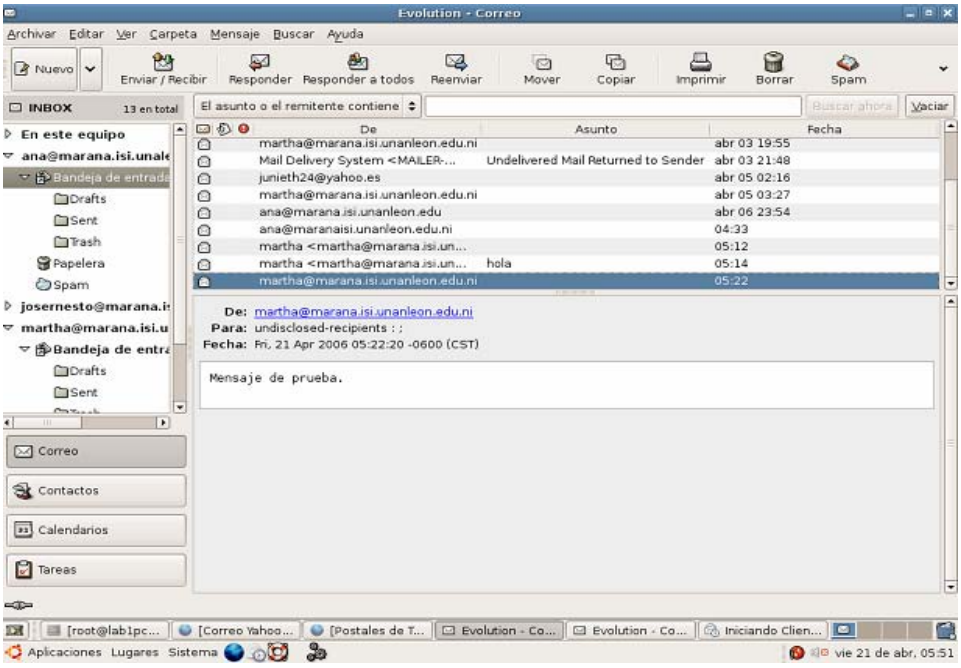

Asistente de cuentas de Evolution
Recepción de correo
Por favor seleccione entre las siguientes opciones
Tipo de servidor: IMAP
Descripción: Para leer y almacenar correo en los servidores IMAP.
Configuración
Servidor: marana.isi.unanleon.edu.ni
Usuario: ana
Seguridad
Usar conexión segura: Nunca
Tipo de autenticación
Contraseña
Comprobar tipos soportados
Recordar contraseña
Cancelar
Atrás
Adelante

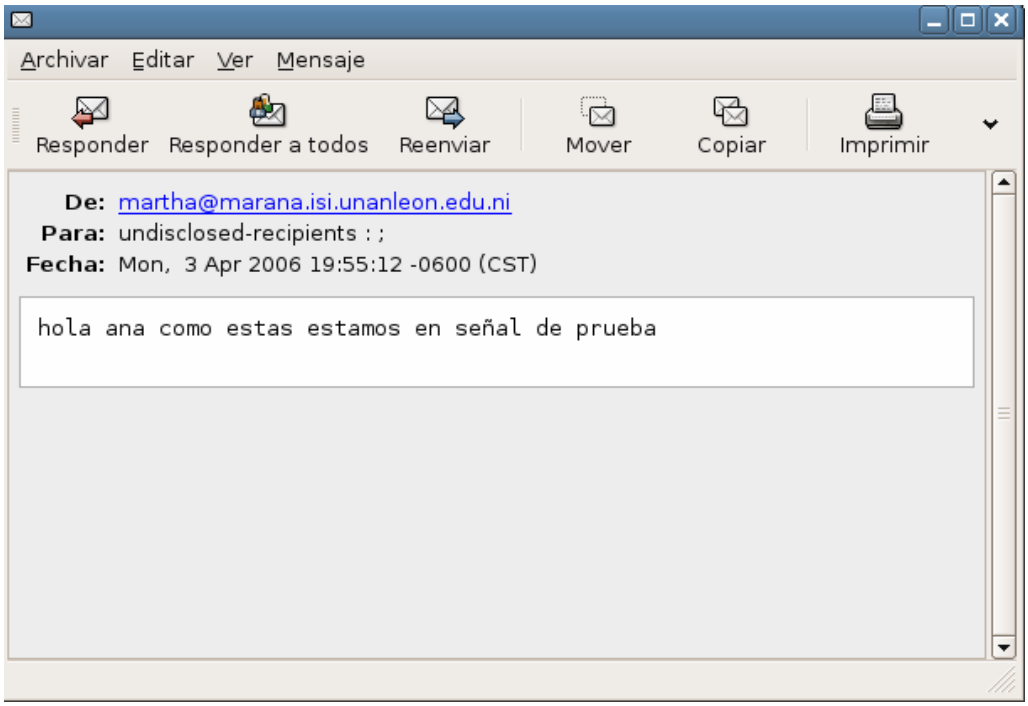

Asistente de cuentas de Evolution
Receiving Options
Comprobación de correo nuevo
Comprobar si hay correo nuevo automáticamente cada 10 minutos
Comprobar si hay mensajes nuevos en todas las carpetas
Conexión con el servidor
Usar un comando personalizado para conectarse al servidor
Comando: ssh -C -l %u %h exec /usr/sbin/imapd
Carpetas
Mostrar sólo las carpetas suscritas
Ignorar el espacio de nombres suministrado por el servidor
Espacio de nombres
Opciones
Aplicar filtros a los mensajes nuevos en INBOX en este servidor
Comprobar si el contenido de los mensajes nuevos es spam
Sólo comprobar si hay spam en la carpeta INBOX
Sincronizar automáticamente el correo remoto localmente
Cancelar
Atrás
Adelante


Asistente de cuentas de Evolution
Envío de correo
Por favor escriba debajo la información acerca de cómo enviará su correo. Si no está seguro, pregúntele a su administrador de sistemas o a su Proveedor de Servicios de Internet.
Tipo de servidor: SMTP
Descripción: Para entregar correo conectándose a un servidor de correo usando SMTP.
Configuración del servidor
Servidor: marana.isi.unanleon.edu.ni
El servidor requiere autenticación
Seguridad
Usar conexión segura: Nunca
Autenticación
Tipo: PLAIN
Comprobar tipos soportados
Usuario: ana
Recordar contraseña
Cancelar
Atrás
Adelante


Asistente de cuentas de Evolution
Administración de la cuenta
Introduzca un nombre descriptivo para esta cuenta en el espacio de abajo.
Este nombre se usará sólo para mostrarlo.
Información de la cuenta
Teclee el nombre por el que quiere identificar a esta cuenta.
Por ejemplo: «Trabajo» o «Personal»
Nombre: ana@marana.isi.unaleon.edu.ni
Cancelar
Atrás
Adelante

Una vez configurada la cuenta puede proceder a leer los mensajes mediante el: Evolution.



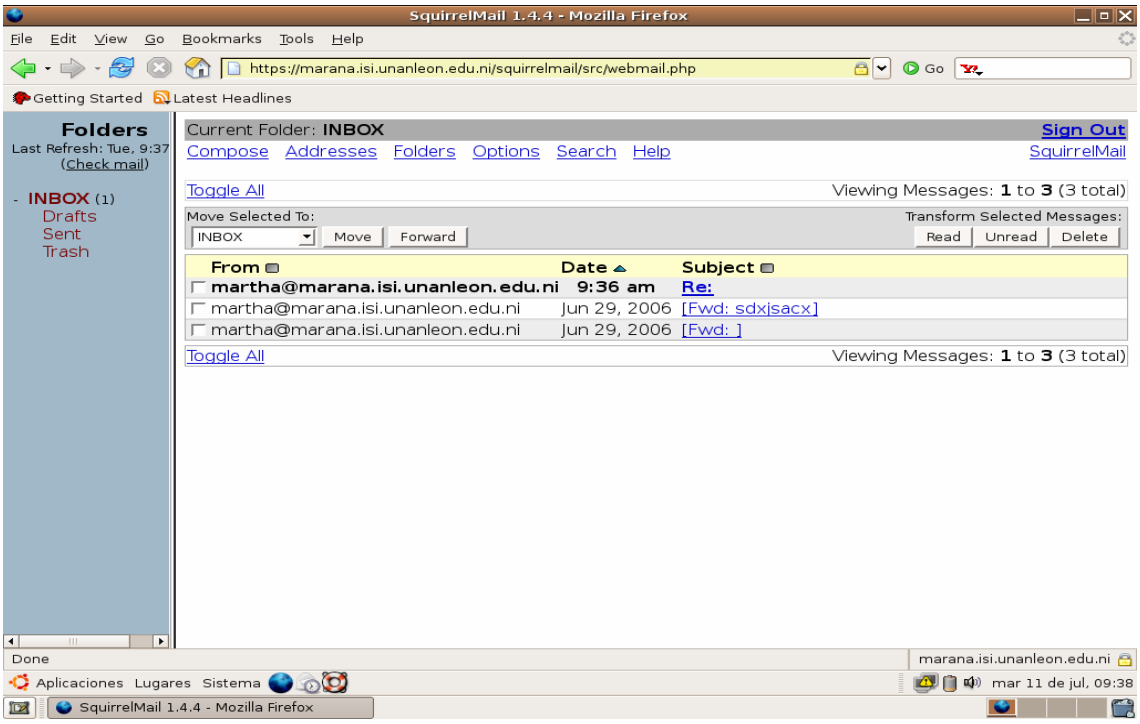


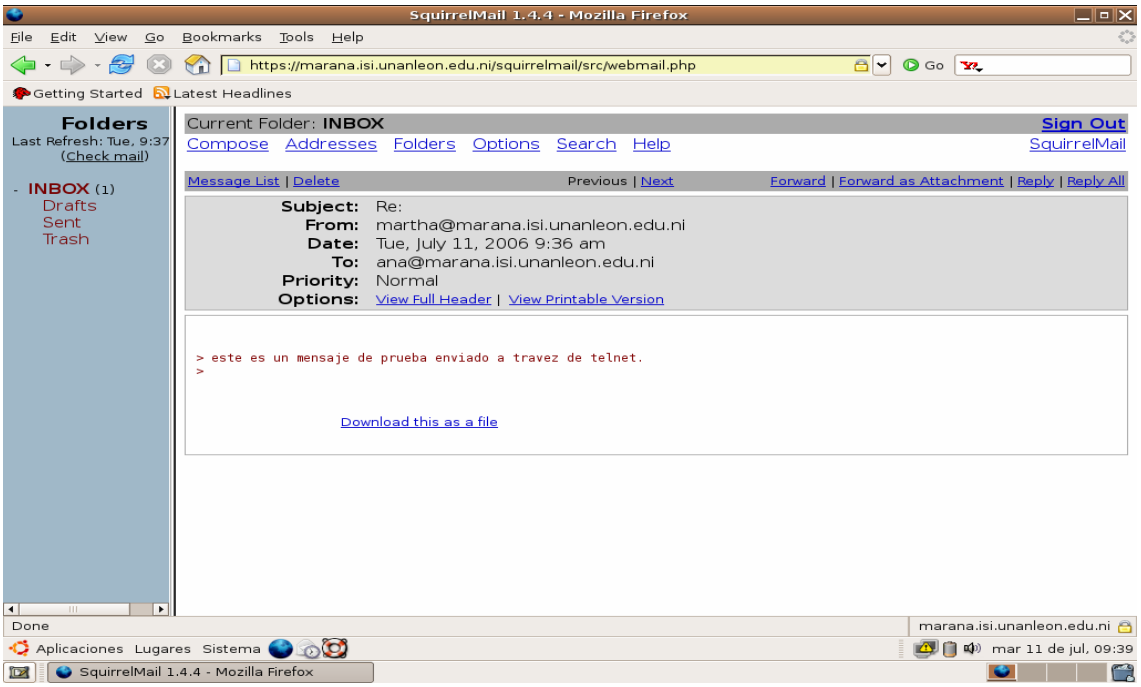
De esta forma ya nuestro servidor puede enviar y recibir mensajes a través del evolution.

Interfaz de squirrelmail con https para revisar el correo.


BIENVENIDOS AL WEBMAIL DE MARANA - Login - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://webmail.marana.isi.unanleon.edu.ni/squirrelmail/src/login.php
Getting Started Latest Headlines
SquirrelMail
webmail
for
nuts
SquirrelMail version 1.4.4
By the SquirrelMail Development Team
BIENVENIDOS AL WEBMAIL DE MARANA Login
Name: ana
Password:
Login


Done
webmail.marana.isi.unanleon.edu.ni
Aplicaciones Lugares Sistema
vie 28 de jul, 09:50
SquirrelMail 1.4.4 - Mozilla Firefox
Mailman Info Page - Mozilla Firefox
BIENVENIDOS AL WEBMAIL DE MARA...


SquirrelMail 1.4.4 - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://marana.isi.unanleon.edu.ni/squirrelmail/src/webmail.php
Getting Started Latest Headlines
Folders
Last Refresh: Tue, 9:37
(Check mail)
INBOX (1)
Drafts
Sent
Trash
Current Folder: INBOX
Sign Out
Compose Addresses Folders Options Search Help
SquirrelMail
Toggle All
Viewing Messages: 1 to 3 (3 total)
Move Selected To:
INBOX
Move
Forward
Transform Selected Messages:
Read
Unread
Delete
From
Date
Subject
martha@marana.isi.unanleon.edu.ni 9:36 am Re:
martha@marana.isi.unanleon.edu.ni Jun 29, 2006 [Fwd: sdxjsacx]
martha@marana.isi.unanleon.edu.ni Jun 29, 2006 [Fwd:]
Toggle All
Viewing Messages: 1 to 3 (3 total)
Done
marana.isi.unanleon.edu.ni
Aplicaciones Lugares Sistema
mar 11 de jul, 09:38
SquirrelMail 1.4.4 - Mozilla Firefox


SquirrelMail 1.4.4 - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://marana.isi.unanleon.edu.ni/squirrelmail/src/webmail.php
Go
Getting Started
Latest Headlines
Folders
Last Refresh: Tue, 9:37
(Check mail)
INBOX (1)
Drafts
Sent
Trash
Current Folder: INBOX
Sign Out
Compose Addresses Folders Options Search Help
SquirrelMail
Message List | Delete
Previous | Next
Forward | Forward as Attachment | Reply | Reply All
Subject: Re:
From: martha@marana.isi.unanleon.edu.ni
Date: Tue, July 11, 2006 9:36 am
To: ana@marana.isi.unanleon.edu.ni
Priority: Normal
Options: View Full Header | View Printable Version
> este es un mensaje de prueba enviado a travez de telnet.
>
Download this as a file
Done
marana.isi.unanleon.edu.ni
Aplicaciones Lugares Sistema
mar 11 de jul, 09:39
SquirrelMail 1.4.4 - Mozilla Firefox