ANEXO E
IMPLEMENTACION DE UN SERVICIO DE CORREO ELECTRÓNICO SEGURO

1. DEFINICIÓN DEL PROBLEMA

Como parte de las necesidades de comunicación y trabajo en red identificadas en el grupo objetivo, se consideró esencial que la Red de Investigación Educativa contara con un servicio institucional de correo electrónico. Esto le permitiría a cada miembro, además de intercambiar mensajes con otras personas dentro y fuera de ieRed, hacer uso de una cuenta que le diera un respaldo y una presencia oficial en Internet, que lo vincularan a esa comunidad académica institucional.

Por otro lado, la cuenta ofrece la posibilidad de albergar mensajes en un espacio de 50MB compartidos con el disco virtual (servicio cuya implementación se explicará en el Anexo F), que representan una cantidad de almacenamiento superior a la que ofrecen la mayoría de las cuentas de correo gratuitas (de 3MB a 5MB) y algunas institucionales (de 7MB a 14MB).

Una vez identificada la necesidad de implementar un servicio de correo electrónico para la Red de Investigación Educativa, se empezaron a examinar un conjunto de alternativas para su prestación, y se determinó que era necesario e imperativo que, independientemente de la selección que se hiciera, se ofreciera el servicio bajo unas condiciones mínimas de seguridad.

Como el concepto de “Servicio de Correo Electrónico Seguro” puede resultar muy ambiguo dependiendo de las consideraciones de seguridad que se tengan en cuenta, más adelante se explicará el conjunto de características mínimas y deseadas de seguridad que a juicio de los autores, debía tener este servicio.

Junto a la World Wide Web, el correo electrónico es quizá uno de los servicios más utilizados por las personas que acceden a Internet, y uno de los más antiguos. El correo electrónico surgió hacia el final de la década de los setentas como un intento de convertir las comunicaciones electrónicas en una nueva forma de intercambiar información entre los centros universitarios más rápida que el correo físico convencional.

Su estructura ha cambiado muy poco desde entonces: el sistema funciona replicando muchos de los elementos del correo ordinario tales como el buzón de correo o dirección postal para recibir mensajes, las agencias postales para el intercambio de correspondencia, las personas encargadas de su entrega final o carteros, un formato estandarizado para el envío de cartas que se compone de los

datos del remitente y los del destinatario escritos en un sobre, y finalmente el mensaje (o carta).

Los primeros (y aún ampliamente utilizados) sistemas para el tratamiento del correo electrónico se concibieron para una Red donde lo que interesaba era compartir, y había poco o ningún espacio para las aplicaciones maliciosas, los virus, los delincuentes o esas indeseables prácticas de publicidad que saturan las buzones electrónicos de las personas. En ese entonces, Internet era un lugar mucho más agradable.

Ante la presencia de éstas y otras amenazas hoy en día, se deben buscar mecanismos para asegurar las comunicaciones y garantizar su privacidad por un lado, y proteger los recursos de las máquinas que soportan el servicio evitando que se haga un mal uso de él, por otro.

Configurar un servicio de una forma completamente segura es quizá algo utópico. No obstante, un sistema se puede considerar relativamente seguro en la medida en que considere un conjunto de primitivas de la seguridad computacional y de redes que se describen a continuación:

- Aceptación: Entendida como la facilidad de brindarle a las personas el acceso a la información. De nada sirve un sistema seguro si nadie puede hacer uso de la información que reside en él.

- Identificación: Se refiere al reconocimiento de un usuario del sistema. Usualmente esto se consigue con un nombre (login) o un número de identificación (id).

- Autenticación: Se refiere a la verificación que se hace de la identidad de un usuario. Incluye desde mecanismos complejos como los sistemas de acceso biométricos[1], hasta los más simples como el uso de contraseñas. En sistemas informáticos, no sólo se refiere al reconocimiento de usuarios, sino también al de procesos que deban comunicarse entre sí en el interior de una máquina.

- Autorización: Consiste en permitirle determinados privilegios y restringirle otros, tanto a un usuario como a un proceso corriendo en un sistema.

- Confidencialidad: Se refiere a una cualidad de la información que la protege de ser accedida por personas, procesos y recursos que no han sido expresamente autorizados para ello.

1 El acceso biómetrico hace referencia a la identificación de una persona a partir de alguna medida física de su cuerpo, como por ejemplo la huella dactilar, el iris, el peso, entre otras.

- Integridad: Consiste en la conservación de la información tanto en tránsito por una red, como la que se encuentra alojada en una localidad de memoria, contra modificaciones de personas no autorizadas para ello o distintas al propietario.

En este orden de Ideas, se concibió un servicio de correo electrónico que tuviese en cuenta aspectos de seguridad tales como:

- Identificación y Autenticación de los usuarios para permitir el envío de correo electrónico sólo aquellos que tuvieran una cuenta válida en el sistema.

- Cifrado de las conexiones entre el cliente y el servidor para evitar que la información que intercambien, incluyendo el contenido del mensaje, pero muy especialmente: el nombre de usuario y la contraseña, se transmitan en texto plano hasta el servidor. Con ello se garantiza la integridad y la confidencialidad de la información en un gran porcentaje, pero hay que aclarar que si el mensaje de correo se enviaba a un servidor externo, la comunicación entre servidores no cuenta con ningún mecanismo de la confidencialidad. Para esta clase de requerimientos, se deben implementar mecanismos que soporten criptografía asimétrica[2] del lado de los clientes.

Estos dos aspectos se debían tener en cuenta tanto en el correo a través de la Web, como a través de un cliente remoto.

Estas características permiten asegurar la disponibilidad del servidor en lo que respecta a fallas debido a exceso de carga en el envío de correo electrónico, evitando que se malgasten los recursos de la máquina y que sea utilizada para enviar correo electrónico no deseado (SPAM).

2. CONCEPTOS BÁSICOS

2.1 Funcionamiento general de un servidor de Correo Electrónico

Un servidor de correo electrónico funciona de forma similar a un enrutador, sólo que en lugar de paquetes, se ocupa exclusivamente del tráfico SMTP (*Simple Mail Transfer Protocol*, Protocolo Simple de Transferencia de Correo). El siguiente es el conjunto de reglas que rige el comportamiento de un servidor SMTP:

2 Un criptosistema asimétrico es aquel en el que cada usuario crea un par de claves, una privada y otra pública. El envío de un mensaje se cifra con la clave pública, y la recepción de un mensaje de descifra con la clave privada. La seguridad del sistema reside en la dificultad computacional de descubrir la clave privada a partir de la pública.(Aguirre 2004).

1. Acepta un mensaje entrante.

2. Comprueba las direcciones del mensaje.

3. Si son direcciones locales, almacena el mensaje para recuperarlo.

4. Si son direcciones remotas, envía el mensaje.

5. Si encuentra que el mensaje no se puede enviar (la cuenta ha excedido su cuota o el usuario ya no existe), devuelve un mensaje de error al remitente que explica el problema.

2.2 Protocolos para el intercambio de Correo Electrónico

Para el intercambio de mensajes entre personas (y archivos adjuntos como imágenes, documentos, de texto, etc.), el servicio de correo electrónico se sirve de diversos protocolos. Estos protocolos permiten que máquinas distintas, que se ejecutan con frecuencia en sistemas operativos y con programas de correo electrónico diferentes, se comuniquen entre sí e intercambien mensajes para que lleguen a los destinatarios adecuados (Red Hat Linux 2002).

Podemos hablar de dos tipos de protocolos: los que le van a permitir a un usuario acceder a su buzón de mensajes en un servidor, y los que le van a permitir enviar mensajes a otros usuarios.

En el primer grupo, los dos protocolos más populares son IMAP (*Internet Message Access Protocol*, Protocolo de Acceso a Mensajes de Internet) y POP (*Post Office Protocol*, Protocolo de Oficina de Correo). La principal diferencia reside en que el protocolo IMAP permite el acceso a los mensajes alojados en el servidor y POP los descarga en la máquina local, borrándolos o dejándo una copia en el servidor, según se indique.

POP fue diseñado inicialmente para leer correos sin conexión. El usuario se conectaba y descargaba los correos a su máquina local después de lo cual éstos eran borrados del servidor. La principal desventaja de esta forma de operación era que no era compatible con el acceso desde múltiples servidores, porque tendía a dispersar el correo por todas las máquinas desde las cuales se revisara. Así, el modo de acceso “sin conexión” ataba a los usuarios a usar un equipo para el almacenamiento y manipulación de mensajes.

IMAP en cambio, fue pensado para permitir el acceso y la gestión de los mensajes desde más de un computador. Además soportaba modos de acceso “en línea”, “sin conexión” y “desconectado”; accesos concurrentes a buzones de correo compartidos; y fue pensado para ser completamente compatible con estándares

de mensajería en Internet como MIME (*Multipurpose Internet Mail Extensions*, Extensiones Multipropósito de Correo en Internet).

En cuanto al segundo grupo, tenemos en él al protocolo SMTP (*Simple Mail Transfer Protocol*, Protocolo simple de transferencia de correo), descrito en el RFC 821.

2.3 Aplicaciones necesarias para el funcionamiento del servicio de correo electrónico

Los sistemas de correo electrónico se componen de varias partes denominadas agentes. Cada agente se responsabiliza de una porción lógica del sistema. Existen cinco agentes: el MUA (*Mail User Agent*, Agente de Usuario de Correo), el MTA (*Mail Transfer Agent*, Agente de Transferencia de Correo), el MDA (*Mail Delivery Agent*, Agente de Entrega de Correo), el MSA (*Mail Submission Agent*, Agente de Registro de Correo), y el MAA (Mail Access Agent, Agente de Acceso al Correo).

El MUA o cliente de correo, es el programa que le va a permitir a un usuario (como mínimo) leer y escribir mensajes de correo electrónico. Típicamente, esto se hace a través de una interfaz que puede ser gráfica (Ximina Evolution, Outlook, Webmail, etc) o en texto (Pine, Mutt, etc). Debe tener funcionalidades de agente de acceso a correo para permitir la recuperación de correo a través de POP o IMAP y debe tener funcionalidad MIME (*Multipurpose Internet Mail Extensions*, Extensiones de Correo de Internet Multipropósito).

La funcionalidad MIME es la habilidad para leer o incluir texto no ASCII (texto plano) en el cuerpo de un mensaje. MIME especifica formas de incluir otra clase de documentos incluyendo imágenes y otros archivos binarios. Esta habilidad depende tanto del MUA como de la existencia de otras aplicaciones capaces de entender el formato del archivo y que puedan ser llamadas o cargadas por el MUA para su visualización.

El MTA se encarga de la transferencia de los mensajes de correo electrónico entre las máquinas que usan el protocolo SMTP. Un mensaje puede pasar por varios MTA hasta llegar al destino final. Los MTA escuchan en los puertos 25 y 587. Típicamente se contactan el uno al otro usando el puerto 25. Los agentes de registro usan el puerto 587. A la transferencia de correo electrónico para un cliente se denomina reenvío (o envío).

Los agentes MTA utilizan programas MDA (*Mail Delivery Agent*, Agente de Entrega de Correo) para entregar el correo electrónico al buzón de un usuario concreto. En muchos casos, el agente MDA es realmente un LDA (*Local Delivery Agent*, Agente de entrega local), como bin/mail o Procmail. Cualquier programa

que gestione realmente un mensaje para entregarlo al punto donde lo leerá un agente MUA se puede considerar un agente MDA. Tenga en cuenta que los agentes MDA no transportan mensajes entre sistemas ni actúan como interfaz para el usuario final.

Muchos usuarios no utilizan directamente agentes MDA, porque sólo se necesitan agentes MTA y MUA para enviar y recibir correo. Sin embargo, algunos agentes MDA se pueden utilizar para ordenar los mensajes antes de que los lea el usuario, lo cual es de gran ayuda si recibe una gran cantidad de correo.

El MSA o Agente de Registro de Correo es un agente nuevo que divide la carga de trabajo del MTA en servicios con muchos usuarios y mejora el desempeño. La idea es que el agente de servicio se preocupe de las tareas relativas al direccionamiento, tomando cierta parte de la carga de trabajo del MTA primario. Éste simplemente puede confiar la validez de las direcciones cuando recibe un correo de agentes de registro conocidos. El MSA corrige direcciones, y arregla y reescribe encabezados. Procesa el correo de su propia cola y lo envía a un agente de transferencia local.

El MAA o Agente de Acceso al Correo es usado para recuperar la el buzón de mensajes de un servidor de correo electrónico. Ejemplos de MAAs son el protocolo IMAP y POP.

En nuestro caso, el MTA que se utilizó en la primera versión del servicio de correo electrónico implementado con el Cliente Web de correo Squirrelmail fue Sendmail. Sendmail desempeña el papel de los agentes de registro y transferencia de correo. Las razones iniciales por las cuales se eligió a este programa son entre otras su gran trayectoria y probada funcionalidad.

No obstante, cuando el área de currículo de RUDECOLOMBIA adquirió el servidor para el soporte de los servicios tecnológicos para la comunicación y el trabajo en ieRed, se cambió a Exim, cuya selección se justificará más adelante.

2.3 El SPAM o correo no deseado

El correo no deseado o SPAM[3] es una de las mayores molestias que deben enfrenar hoy en día tanto los usuarios como los administradores del servicio de correo electrónico. La cantidad de “correo basura” que puede inundar los buzones de correo puede causar desde improductividad por el tiempo que consumen las personas tratando de descriminar qué es SPAM y qué no lo es, hasta congestión en los servidores de correo electrónico por la cantidad de mensajes que deben

3 El origen de la palabra aún es incierto, pero varios autores arguyen que significa: *Sales Promotional/Advertising Mail*, Correo Promocional/Publicitario para Ventas

procesar. La mayor parte del SPAM se transmite de forma masiva, lo que aumenta el nivel de congestión en las redes que emplea para su transporte. Este último es el caso que más preocupaba a los autores.

2.4 Mecanismos de Autenticación

Dentro de los mecanismos de autenticación disponibles, los más comunes son: PLAIN, LOGIN, CRAM-MD5[4], DIGEST-MD5, y NTLM (NT LAN Manager). De éstos, PLAIN, CRAM-MD5, y DIGEST-MD5 son mecanismos de autenticación estandarizados, mientras que LOGIN y NTLM son mecanismos propietarios de Microsoft. Sólo PLAIN y LOGIN puede utilizar la contraseña de usuarios en sistemas Unix o Linux.

Como se puede observar en la documentación existente para SASL 0.1[5]: el uso de los diferentes mecanismos de autenticación, depende de los requerimientos de la aplicación que los esté usando. Mecanismos simples como LOGIN y PLAIN, están dirigidos a anclarse en mecanismos de autenticación existentes tales como / etc/passwd a través de PAM (*Pluggable Authentication Module*, Módulo de Autenticación Conectable). La respuesta del cliente a estos mecanismos es sencilla de implementar: al usuario sólo se le pide su nombre de usuario y su contraseña, y luego las llamadas al servidor pasan el nombre de usuario y la contraseña a las políticas de decisión definidas por el sistema de autenticación.

En otros mecanismos como CRAM-MD5 y su sucesor, DIGEST-MD5, la autenticación se basa en secuencias de desafío-respuesta y reposa en la posesión de un “secreto” de cierto tipo para efectuar la autenticación. El servidor genera un desafío y el cliente le responde probando que el conoce el secreto, es decir, la respuesta al desafío. Típicamente este “secreto”, es una contraseña generada a partir de algoritmos de resumen (hash). La respuesta del cliente es la misma que para PLAIN o LOGIN. Como sea, el servidor no recibe la contraseña en texto plano a través de la red sino un resumen de ésta. Dado que los sistemas de autenticación de políticas de decisión como PAM no los pueden manipular, la respuesta del servidor para estos mecanismos es más complicada.

A continuación se describen en forma breve los mecanismos de autenticación más usados:

4 MD5 (*Message-Digest Algorithm 5*, Algoritmo de Resumen de Mensaje 5) es un tipo de algoritmo de resumen de mensajes (y una función hash criptográfica) con un valor hash de 128 bits. Fue diseñado por Ronald Rivest del MIT (*Massachusetts Institute of Technology*, Instituto Tecnológico de Massachusetts). Generalmente las sumas de MD5 se codifican como un número hexadecimal de 32 dígitos.

5 Disponible en Internet en: http://www.gnu.org/software/gsasl/manual/gsasl.html#Mechanisms

- PLAIN emplea un nombre de usuario (identidad de autenticación e identidad de autorización) y una contraseña para autenticarlos. Proporciona dos formas de validar al usuario, bien sea recuperando la contraseña bruta de la aplicación con el mecanismo SASL, o llamando a la aplicación con la identidad de autenticación, la identidad de autorización y la contraseña, y permitiéndole decidir entre éstas.

- LOGIN utiliza el nombre de usuario (únicamente la identidad de autorización) y la contraseña para autenticarlos. Se proporcionan dos formas de validar el usuario, bien sea haciendo que el mecanismo SASL recupere la contraseña en bruto de la aplicación y efectúe la validación internamente, o llamando a la aplicación, y permitiéndole decidir entre la identidad y la contraseña de autorización. Si la aplicación especifica tanto las llamadas de validación como de recuperación de contraseñas, se usará la de validación.

- CRAM-MD5 utiliza el nombre del usuario (Sólo la identidad de autorización) y la contraseña para autenticar a los usuarios. Únicamente transfiere la contraseña resumida, lo que significa que no se pueden usar sistemas convencionales de autenticación de políticas como PAM, porque éste no soporta la extracción de contraseñas. CRAM-MD5 proporciona dos formas de validar al usuario: bien sea haciendo que el mecanismo SASL recupere la contraseña en bruto de la aplicación y efectúe la validación internamente, o llamando a la aplicación con el desafío y la respuesta CRAM-MD5, y permitiéndole decidir. Si se especifica tanto la validación como la llamada para recuperar contraseñas, se usará la primera.

- El mecanismo DIGEST-MD5 se basa en la misma operación criptográfica de CRAM-MD5, pero soporta otras características como la identidad de autorización (autenticación proxy) y la protección criptográfica de los datos. Al igual que CRAM-MD5, sólo transfiere la contraseña resumida, lo que implica que no se pueda usar, por ejemplo, PAM como plataforma de transporte, dado que ésta no soporta la extracción de contraseñas. DIGEST-MD5 proporciona dos formas de validar al usuario: una haciendo que el mecanismo SASL recupere la contraseña en bruto de la aplicación y efectúe la validación internamente, y la otra, haciendo que el mecanismo SASL recupere la la versión resumida de la contraseña. La ventaja de usar esta última es que no es necesario guardar las contraseñas de usuario en texto plano en el servidor, sino un resumen unidireccional de éstas, con el nombre de usuario y el dominio. Aún así, el resumen unidireccional del secreto debería manejarse de la misma forma que una contraseña en texto plano. La ventaja está en que si alguien roba el resumen unidireccional, no podrá leer la contraseña del usuario inmediatamente. Si la aplicación especifica ambas llamadas, se usará la que recupera el resumen secreto.

- NTLM emplea el nombre de usuario (la identidad de autorización solamente) y la contraseña para autenticar a los usuarios. Sólo el lado del cliente es implementado.

3. ALTERNATIVAS DE SOLUCIÓN CONSIDERADAS Y JUSTIFICACIÓN DE LAS SOLUCIÓN ESCOGIDA

Dentro de las alternativas consideradas para la implementación de un servidor de correo electrónico seguro, se debían examinar varios aspectos, que se podrían resumir de la siguiente manera: selección del MTA, selección del mecanismo de autenticación de los usuarios para el control del reenvío de correo, y finalmente, selección del cliente Web de correo electrónico.

3.1 Selección del MTA

Se consideraron tres alternativas de forma teórica: Sendmail, Qmail y Exim; y dos en forma práctica: Sendmail y Exim; que fueron instalados, configurados y puestos en funcionamiento durante un tiempo, bajo condiciones reales de operación, aunque eso sí, con una carga y una exigencia muy bajas en el servicio, debido a la poca cantidad de mensajes que tuvieron que procesar.

El trabajo teórico consistió esencialmente en una exploración sobre el nivel de utilización de cada MTA en Internet, la cantidad de proyectos vinculados, una lectura de sus características, forma de operación, extensiones de seguridad, historial de seguridad, madurez de sus proyectos, y modo de configuración, etc.

En esta exploración se encontró que Sendmail seguía siendo el MTA más popular, y el preferido a la hora de buscar características avanzadas en el tratamiento del servicio de correo. El principal problema del pleno aprovechamiento de todo su potencial, era el profundo conocimiento que había que tener de la estructura del programa y de su lenguaje de configuración, para sacar provecho de ellas, lo que hacía relativamente tedioso y demorado su proceso de configuración. Por otro lado, la estructura de Sendmail es monolítica y cualquier nueva funcionalidad que se desease agregar, requería su recompilación. Sendmail tiene un historial de seguridad bastante conocido, pero la complejidad de su configuración a menudo lo hacen complicado de asegurar.

La siguiente alternativa considerada fue Qmail, el MTA de Dan Bernstein, desarrollado con los problemas de seguridad más graves de Sendmail en mente y con la idea de que la seguridad no fuese la meta, sino el punto de partida. Qmail es famoso por este enfoque, por su modularidad y porque el autor ofrece $1000 U.S. a quien encuentre un fallo de seguridad en su código. En varios lugares del

sitio Web oficial[6], se le promociona como “el segundo MTA más usado del mundo”. Existe abundante documentación a su alrededor y aparentemente es más fácil de configurar y más flexible que Sendmail.

Finalmente se encuentra Exim, el MTA desarrollado por Philip Hazel de la Universidad de Cambridge, para ser usado en sistemas Unix/Linux conectados a Internet, y con una fuerte preocupación por lograr una codificación impecable en su construcción. Su diseño original era Similar al de Smail 3, pero con más funcionalidades. Exim proporciona mecanismos para el control de la proliferación de SPAM, y protección contra los bombardeos de correo electrónico (mail bombing). Es además un MTA completamente libre[7], a diferencia de Qmail, que en el mejor de los casos, son sólo de Código Abierto (Open Source).

Existían otras alternativas tales como Smail, que fue descartado por considerarse un proyecto con poco avance, a pesar de haber sido el primer intento serio de construir un reemplazo para Sendmail; y Postfix, porque a pesar de ser uno de los principales contendientes de Qmail, no presentaba ventajas claras frente a éste.

En diferentes estudios sobre la utilización de MTAs en Internet, se encontró que de las alternativas consideradas, efectivamente Sendmail era el que tenía una mayor acogida, seguido de Qmail y Exim, intercalándose con Postfix dependiendo de la fuente consultada.

Después de examinar la alternativas existentes, el MTA seleccionado por varias razones, entre las que sobresalen su fuerte orientación a la protección y el control del envío de correo no deseado, y su fuerte vinculación al Proyecto Debian, fue Exim. Exim es el MTA de facto para la distribución Debian GNU / Linux, sugerido en la instalación y configuración del sistema operativo y sus aplicaciones. En este momento, Exim es un proyecto maduro, con muchos logros y soporte para diversas extensiones, tales como SMTP STARTTLS/AUTH, esenciales en el servicio de correo electrónico a implementar y que se explicarán más adelante. La versión de Exim implementada, fue la 4.30-4 disponible para Debian Sarge.

3.2 Selección del mecanismo de autenticación para el control del envío de correo

Para controlar el envío de correo electrónico desde un servidor, el problema se puede abordar desde dos perspectivas. La primera de ellas es ejercer un control sobre las direcciones de los destinatarios del mensaje, que puede ser de poca utilidad porque se restringe la libertad de los usuarios de enviar mensajes a donde

6 http://www.qmail.org/top.html

7 Se distribuye bajo la GNU GPL (GNU General Public License, Licencia Pública General de GNU)

quieran, a un conjunto limitado de destinos; la segunda es filtrar el correo electrónico según su origen.

Mientras la primera forma es fácil de controlar, la segunda se puede convertir en un problema: ¿Cuál es el origen de una dirección de correo? Es posible basarse en la dirección IP o nombre del host, pero si se quiere determinar el origen de una solicitud para envío de correo a nivel de usuario, las cosas se complican.

Con la restricción a direcciones de correo locales para el envío de mensajes, se evita que el servidor sea usado para enviar SPAM a otros destinos, pero no se evita que los usuarios locales lo reciban (en el caso de que el SPAMMER se encuentre dentro de la red local).

Además, esta alternativa deja por fuera la posibilidad de que un usuario, que se conecte desde un cliente de correo remoto que no le permita acceso directo a la máquina, pueda enviar mensajes. Es decir que si se tienen usuarios empleando programas como el Ximian Evolution o el Microsoft Outlook, a través de una conexión por MODEM a un ISP (*Internet Service Provider*, Proveedor de Servicio de Internet), no van a poder enviar mensajes desde el servidor, a menos que se habilite el reenvío abierto, con lo que se abre la puerta para que cualquiera use el servidor como fuente de SPAM, contribuyendo al crecimiento de este flagelo, y comprometiendo los valiosos recursos de la máquina.

Si se usa Sendmail como MTA, este tipo de control se logra modificando el archivo de control de transmisiones (generalmente /etc/mail/access) de modo que permita únicamente las transmisiones de los hosts o grupos de hosts autorizados y rechace cualquier otra petición. En Exim 4, se consigue definiendo listas de control de acceso en /etc/exim4/conf.d/acl

Existe la posibilidad de personalizar el mensaje de error que emite el servidor cuando se rechazan las peticiones de envío, e incluso no emitir ningún mensaje. Con esto se evita que los eventuales spammers reciban alguna información sobre el motivo por el cual no fue exitosa su petición de conexión al puerto 25 al servidor.

De esta forma, se definieron un conjunto de criterios para enfrentar el problema de permitirle un envío controlado a los usuarios que se conectaran en forma remota a través de clientes de correo y se definieron el siguiente conjunto de criterios para su implementación:

- La aplicación escogida debía ser fácil de implementar en el servidor de correo, fácil de configurar y sobre todo, debía ser compatible el MTA que se seleccionase para operar en el servidor SMTP.

- Debido a la escasa experiencia en el uso de programas computacionales de la mayoría de los usuarios del servicio de correo electrónico a implementar, la aplicación escogida debe ser fácil de configurar en el cliente de correo electrónico (MUA).

Así, se encontró que existían esencialmente dos formas de autenticar usuarios con cuenta de correo en una máquina para permitirles el envío de correo: Mediante técnicas alternativas o "hacks", o implementando SMTP STARTTLS/AUTH.

Las técnicas alternativas o "hacks", proveen formas menos sofisticadas de autenticación que no utilizan autenticación SMTP. La mejor de estas técnicas es *POP antes de SMTP*.

El envío autorizado de correo basado en la autenticación provista por un demonio POP modificado (Pop antes de SMTP), permite que usuarios con una cuenta válida en el sistema puedan enviar mensajes desde un servidor de correo electrónico, si este permite recuperación de los mensajes en el servidor a través de los protocolos POP o IMAP.

Con el fin de que la autenticación de los usuarios se haga en forma segura, los demonios POP o IMAP que corran en el servidor deben ser preferiblemente POPS e IMAPS (POP e IMAP seguro o sobre SSL/TLS[8]).

El procedimiento que sigue esta técnica es sencillo: para enviar mensajes, el usuario primero debe proporcionar su contraseña de acceso al buzón de correo en el servidor y una vez validado, se le permitirá el reenvío por un espacio de tiempo determinado (se recomienda que éste no sea inferior a 5 minutos ni superior a 1 hora), después del cual tendrá que revalidarse si quiere seguir enviando mensajes. Esto se consigue obteniendo la dirección IP del usuario a través de la validación que realiza POP, permitiendo el envío de correo desde esa IP por un tiempo limitado.

Pop antes de SMTP es una idea de John Levine descrita por Scott Hazen Mueller e implementada por Neil Harkins y John Levine. Para implementarlo, se requieren algunas modificaciones al demonio POP, algunas utilidades, y una adición sencilla a la configuración del MTA.

En la primera implementación que se hizo del servicio de correo electrónico para la Red de Investigación Educativa, y en la cual se utilizó Squirrelmail como cliente Web de correo, se utilizó esta técnica debido a su facilidad de instalación y

8 TLS (*Transport Layer Security*, Seguridad de la capa de Transporte) y SSL (*Secure Sockets Layer,* Capa de Conectores Seguros).

configuración, en comparación con el trabajo que suponía implementar SMTP AUTH/STARTTLS con Sendmail[9]. La utilidad empleada fue Poprelay[10].

SMTP STARTTLS (Extensión de inicio de TLS/SSL para SMTP) es el comando para iniciar la seguridad de la capa de transporte (*Start Transport Layer Security*) o en otras palabras, activar SSL.

Según el RFC 2487, la extensión STARTTLS para SMTP se conforma de la siguiente manera:

- El nombre del servicio SMTP definido es STARTTLS;
- la valor de la clave EHLO asociada con la extensión es STARTTLS;
- esta clave no tiene parámetros;
- se define un nuevo verbo SMTP: "STARTTLS";
- no se suman parámetros adicionales a ningún comando SMTP.

La clave STARTTLS es usada para decirle al cliente SMTP que el servidor SMTP permite el uso de TLS.

El comando STARTTLS es: *STARTTLS*, sin ningún parámetro. Después de que el cliente proporcione el comando STARTTLS, el servidor le responderá con alguno de los siguientes códigos de respuesta:

220 Listo para iniciar TLS

501 Error de sintaxis (no se permite ningún parámetro)

454 TLS no se encuentra disponible debido a una razón temporal

Un servidor SMTP referenciado públicamente no debe solicitar el uso de la extensión STARTTLS para distribuir el correo localmente. Esta regla previene que la extensión SMTP dañe la interoperabilidad de la infraestructura SMTP de Internet. Un servidor SMTP referenciado públicamente es un servidor SMTP que corre en el puerto 25 de un host de Internet listado en un registro MX (Mail eXchange, Intercambio de Correo), o un registro A, si no hay registros MX

9 Para mayor información se puede visitar: http://www.sendmail.org/~ca/email/auth.html y http://www.sendmail.org/~ca/email/starttls.html

10 La sitio Web de este proyecto es: http://poprelay.sourceforge.net/

presentes, para el nombre de dominio a la derecha de la dirección de Correo de Internet.

TLS puede proporcionar autenticación (identificación de las partes participantes en la comunicación), privacidad/confidencialidad (la comunicación no es interceptada o husmeada), e integridad (el mensaje no ha sido modificado). Emplea diferentes algoritmos para la encripción, firma, autenticación de mensajes, etc.

STARTTLS puede ser usado para permitir el envío de correo basado en certificados, y para restringir conexiones entrantes o salientes. Para este propósito, hay disponibles diversos conjuntos de reglas que requieren algunos macros nuevos (tales como el tramitador del certificado, el asunto del certificado, la versión de TLS/SSL usada, etc) y el mapa de acceso (que permite definir el acceso al sistema mediante la verificación de dominios y direcciones de correo electrónico para aceptar, rechazar y enviar mensajes).

Para usar un MTA con STARTTLS como servidor y como cliente, se necesita obtener e instalar uno o varios certificados de una CA (CA *Certificate Autorithy*, Autoridad de Certificado) y modificar el archivo de configuración que competa para ello.

Estos certificados de una Autoridad de Certificado se necesitan para autenticar satisfactoriamente a otra entidad. La firma del certificado presentado por la contraparte es verificada a través de estas Autoridades de Certificados. Si una de ellas emitió el certificado, la autenticación es considerada exitosa. Es más, durante el “apretón de manos” (handshake), los DN (*Distinguished Names*, Nombres Distinguidos) de estos certificados son enviados al cliente de tal forma que pueda seleccionar apropiadamente el certificado que está firmado por una de las CA.

Ventajas de STARTTLS

- Autenticación: el cliente y el servidor de una conexión SMTP pueden ser identificados.

- Privacidad/confidencialidad: la transmisión de un correo electrónico entre un cliente y el servidor utilizando STARTTLS no puede ser leída y retraducida si se ha provisto y negociado un paquete de cifrado lo suficientemente seguro.

- Integridad: El texto plano de un correo electrónico entre un cliente y un servidor utilizando STARTTLS no puede ser modificado por un adversario si se ha provisto y negociado un paquete de cifrado lo suficientemente seguro.

Limitaciones de STARTTLS

Todas estas ventajas son provistas transparentemente por los MTAs sin interacción con los usuarios. Estos no necesitan tener un software especial instalado en sus MUAs que sea adicionalmente compatible con el software del recipiente. Esta es al tiempo, la razón de varias limitaciones:

- No proporciona una encripción punto a punto, por lo que usualmente, un usuario no podrá controlar la transmisión completa. Esto contrasta con el uso de TLS por HTTP (*HiperText Transfer Protocol*, Protocolo de Transferencia de Hiper Texto): aquí el cliente del usuario (un navegador Web) se conecta directamente al servidor que provee los datos. El correo electrónico puede ser transferido a través de múltiples saltos de los que el remitente puede controlar al menos el primero.

- No proporciona autenticación de mensajes, a menos que el email haya sido enviado directamente desde el MUA del cliente (con soporte para STARTTLS) a los MTA receptores que deben grabar el certificado del cliente. Aún entonces el mensaje podría ser falsificado durante el reparto local.

En suma, para obtener privacidad, integridad y autenticación punto a punto entre los usuarios se debe usar un software como PGP (*Pretty Good Privacy*, Privacidad muy buena) o S/MIME (*Secure / Multipurpose Internet Mail Extensions*, Extensiones Multipropósito de Correo de Internet Seguras). Esto requiere por lo menos de cierto conocimiento de tal software y un uso responsable de los usuarios finales.

SMTP AUTH (SMTP *AUTHentication*, Autenticación SMTP) es una extensión para el servicio de autenticación del protocolo SMTP.

Consultando el RFC 2554, se encontró lo siguiente:

- El nombre de la extensión del servicio SMTP es "Autenticación";

- el valor de la clave EHLO asociada con la extensión es "AUTH".

- La clave AUTH EHLO contiene como parámetro una lista de los nombres de los mecanismos SASL (Simple Authentication and Security Layer, Capa de Autenticación y Seguridad Simple) soportados, separados por espacios.

- Se define un nuevo verbo SMTP: "AUTH";

- se adiciona un parámetro adicional empleando la clave "AUTH" al comando MAIL FROM, y se extiende el número máximo de líneas del comando MAIL FROM en 500 caracteres.

- esta extensión es apropiada para el protocolo de registro (sumisión o envío) [SUBMIT].

El comando AUTH indica un mecanismo de autenticación al servidor. Si el servidor soporta el mecanismo de autenticación solicitado, efectúa un protocolo de intercambio de autenticación para autenticar e identificar al usuario. Opcionalmente, también negocia una capa de seguridad para interacciones de protocolo subsecuentes. Si el mecanismo de autenticación no se encuentra soportado, el servidor rechaza el comando AUTH con una respuesta 504.

El protocolo de intercambio de autenticación consiste en una serie de desafíos del servidor y respuestas del cliente que son específicas del mecanismo de autenticación.

Al servidor no se le pide que soporte un mecanismo de autenticación específico, ni se le pide a los mecanismos de autenticación que soporten ninguna capa de seguridad. Si un comando AUTH falla, el cliente puede intentar otro mecanismo de autenticación empleando otro comando AUTH.

Si un cliente emplea esta extensión para obtener un túnel encriptado hacia un servidor a través de una red insegura, necesita configurarse para nunca enviar correo al servidor cuando la conexión no esté mutuamente autenticada y cifrada. De otra parte, un atacante podría robar el correo del cliente suplantando la conexión SMTP, o pretendiendo que el servidor no soporta la extensión de autenticación, o causando que todos los comandos AUTH fallen.

Antes de que la negociación SASL haya comenzado, cualquier interacción del protocolo es realizada en forma desprotegida y podría ser modificada por un atacante activo. Por esta razón, los clientes y los servidores deben descartar cualquier conocimiento obtenido previamente del otro antes del inicio de la negociación SASL en cumplimiento de la negociación (SASL), que resultará en una capa de seguridad.

Este mecanismo no protege el puerto TCP, así que un atacante activo puede redirigir un intento de conexión de envío al puerto al protocolo de registro [SUBMIT]. El parámetro AUTH=<> previene un ataque así de causar el envío de un mensaje sin un envoltorio de identificación para recoger la autenticación del cliente de envío.

Un cliente de registro de mensaje puede solicitar al usuario que se autentique cuando quiera que se informe sobre la disponibilidad de un mecanismo compatible con SASL. Por ello, puede no ser deseable para un servidor de registro [SUBMIT] informar de la existencia de un mecanismo SASL cuando el uso de ese mecanismo no le garantice ningún beneficio al cliente sobre un registro anónimo.

Esta extensión no pretende reemplazar o ser usada en el lugar de los sistemas de firma y cifrado de mensajes punto a punto tales como S/MIME o PGP. Esta extensión aborda un problema diferente a los sistemas punto a punto; tiene las siguientes diferencias claves:

- Es útil generalmente sólo dentro de un enclave confiable;
- Protege todo el envoltorio del mensaje, no sólo el cuerpo del mensaje;
- Autentica el registro del mensaje, no la autoría del contenido del mensaje;
- Puede darle al remitente cierta seguridad de que el mensaje fue enviado al siguiente salto en los casos en que el remitente se autentica mutuamente con el siguiente salto y negocia una capa de seguridad apropiada.

SASL define dos términos que son importantes en este contexto: identificador de autorización (userid), que es utilizado por las aplicaciones para verificar qué operaciones están permitidas (autorizadas), y el identificador de autenticación (authid), que se utiliza para autenticar al cliente, es decir que las credenciales de autenticación para el cliente contienen el identificador de autenticación. Este puede ser utilizado por un servidor proxy para actuar como otro usuario.

SMTP AUTH permite el envío de remitentes que se hayan autenticado satisfactoriamente. Por defecto, el reenvío está permitido para cualquier usuario que se haya autenticado a través de un mecanismo confiable, esto es, uno que esté definido a través de TRUST_AUTH_MECH (“lista de mecanismos confiables”). Este mecanismo es útil para usuarios móviles y puede reemplazar la técnica alternativa POP antes de SMTP si el MUA soporta SMTP AUTH.

No se recomienda la utilización de PLAIN ni LOGIN como mecanismos de autenticación, a menos que tenga activa una fuerte capa de encripción, como STARTTLS o un túnel SSL externo. Esta es la razón por la que se habla de SMTP STATTLS/AUTH. Son mecanismos que se complementan para permitir opciones tanto de autenticación como de encriptacíon en un servicio de correo electrónico.

En resumen: cierto nivel de solapamiento entre los dos estándares permitir autenticar clientes TLS mediante certificados, utilizar estos certificados para autenticar máquinas, renegociar un enlace seguro mediante SASL, y controlar el permiso de envío a los usuarios (mediante la solicitud de nombres de usuarios y contraseñas).

3.3 Selección del software para el soporte del protocolo IMAP

El protocolo IMAP (cuya especificación figura en el RFC 3501) es esencial para soportar un servicio de correo electrónico completo que permita que clientes remotos se conecten a través de un MUA. Para su implementación, existen múltiples alternativas como Courier IMAP, Cyrus IMAP, UW-IMAP, entre otras[11].

De las opciones posibles, se escogió UW-IMAP (*University of Washington IMAP Toolkit*). Esta alternativa era conocida debido al trabajo realizado en la primera implementación del servicio de correo electrónico con Sendmail sobre un Red Hat Linux 9.0, y había dado excelentes resultados.

Esta implementación del protocolo utiliza mbox, que es el formato de almacenamiento tradicional en máquinas Unix/Linux. En la documentación existente sobre UW-IMAP, se explica que no utiliza el formato maildir debido a las dificultades técnicas que encierra soportarlo, mientras se mantiene un desempeño, una robustez y se siguen tanto los requerimientos del protocolo IMAP como del formato maildir, de manera simultánea.

Por otro lado, aunque existen múltiples estudios sobre el desempeño de ambos tipos de formato, todos tienden a mostrar una ligera superioridad en el desempeño usando el formato mbox que el maildir para volúmenes elevados de mensajes (más de 10000).

La versión de UW-IMAP implementada fue la 2001a-debian-6, disponible para Debian Woody 3.0r2. Esta permite tanto conexiones de IMAP seguro (puerto 993), como de IMAP inseguro (puerto 143).

La razón por las que se requería una conexión al puerto no seguro del servicio, es porque el cliente web de correo electrónico (Openwebmail), así lo requería. No obstante, dado que el servicio de acceso al correo web se implementó empleando SSL, la transferencia de correo y el intercambio del nombre de usuario y la contraseña con el servidor se hacían de manera segura.

De esta forma, se tenían conexiones inseguras entre la aplicación Openwebmail y UW-IMAP localmente en el servidor, y conexiones IMAP seguras para los clientes que se conectaran a través de clientes de correo como Ximian Evolution, Kmail o Microsoft Outlook.

11 Para conocer una lista más completa de las implementaciones de este protocolo, puede visitarse: http://www.imap.org/products/showall.php

3.3 Selección del cliente Web de correo electrónico

Para la selección del cliente Web de correo electrónico, se tuvieron en cuenta un conjunto de criterios, que se pueden resumir a continuación:

- Compatibilidad con el MTA seleccionado.
- Soporte para cambio de la contraseña sin necesidad de incluir componentes adicionales.
- La aplicación debía ser software libre.
- La aplicación debía ser usable, en especial en aspectos relacionados con las funcionalidades básicas (ingresar al correo, enviar y recibir mensajes, adjuntar archivos, cambiar la contraseña, etc.)
- La aplicación debía ser compatible con una implementación segura del Servidor Web.
- La aplicación debía proporcionar un mecanismo de acceso vía Web al servicio de Disco Virtual, que se explicará con mayor detalle en el anexo F.

Estas y otras características, se evaluaron a través de un análisis heurístico de Usabilidad, es decir, guiado por un conjunto de principios que se consideran pautas muy acostumbradas a seguir en el diseño de interfaces Web.

En general, el examen de la aplicación giró en torno a un conjunto de criterios derivados de principios bien conocidos en el diseño Web, que los autores agruparon en:

- Diseño Gráfico: Claridad en el lenguaje visual, representaciones gráficas comprensibles, colores y enlaces estándar, y distribución adecuada de los elementos.
- Percepción: La percepción de la interfaz se refiere al nivel de contextualización y conciencia que ésta es capaz de transmitirle al usuario. Reúne elementos como: esquemas adecuados de navegación y búsqueda, ubicación dentro de la aplicación, e información sobre el estado del sistema y sobre los errores.
- Funcionalidad: Se refiere a la capacidad de distinguir y utilizar las opciones de la aplicación. La interfaz debe facilitar la ubicación, identificación y disponibilidad de las funciones del sistema; así como el pleno control sobre las operaciones.

Las aplicaciones tenidas en cuenta a lo largo del proyecto para la implementación del cliente Web de correo electrónico fueron básicamente: Ilohamail (http://ilohamail.org/), Horde IMP (http://horde.org/imp/), Squirrelmail (http://squirrelmail.org/), y Openwebmail (http://openwebmail.org/). Aparte de éstas, se examinaron otras que fueron descartadas por la poca madurez de sus proyectos, la falta de características claves, y la ausencia de una interfaz atractiva y fácil de usar. Las tres primeras fueron instaladas, configuradas y sometidas a pruebas de usabilidad con usuarios reales. En esta primera selección, la más opcionada fue fue Squirrelmail, y se montó con Sendmail como MTA.

Efectuando una segunda exploración para los servicios que debían migrarse de Red Hat Linux 9.0 a Debian 3.1, se confrontó mediante análisis heurístico a Squirrelmail con Openwebmail, y finalmente se optó por ésta última. Las razones para esta elección se debieron en parte a su soporte del Disco Virtual, la sencillez de la interfaz, la calidad de los textos y contenidos que traía para el idioma español, la claridad en los mensajes de error que generaba el programa si sucedía algo, la facilidad con que podía realizarse su configuración, entre otras.

Openwebmail está orientado al soporte del acceso Web a archivos de correo de gran tamaño, de una manera eficiente. Está escrito en Perl (*Practical Extraction and Report Language*, Lenguaje Práctico de Extracción y Reporte)[12], un lenguaje estable, independiente de la plataforma, usado en proyectos de misión crítica y en aplicaciones Web multipropósito.

Dentro de las características encontradas en la documentación de éste cliente Web de correo, sobresalen:

- Acceso rápido a las carpetas de mensajes.
- Traslado eficiente de mensajes
- Menor memoria utilizada que otros clientes Web de correo
- Interfaz gráfica agradable
- Permite el envío SMTP remoto
- Soporte para hosts virtuales
- Soporte para alias de los usuarios
- Amplias opciones de configuración para el usuario

12 http://www.perl.org/

- Soporte para varios métodos de autenticación
- Soporte para PAM
- Opción de búsqueda en todo el contenido
- Fuerte soporte para MIME (en presentación y composición)
- Soporte para carpeta de borradores
- Respuesta a correos a través de utilidades en la interfaz
- Incluye una opción muy completa para hacer corrección ortográfica
- Soporte para filtros de mensajes
- Previsualización de cantidad de mensajes existentes y procesados
- Conversión de codificación de caracteres automática
- Soporte para Disco Web (Virtual)
- Corre en forma persistente a través de SpeedyCGI (Una utilidad para incrementar la velocidad de scripts en perl corriéndolos de manera persistente).
- Soporte para compresión HTTP

4. IMPLEMENTACIÓN DE LA SOLUCIÓN

Para implementar el servicio de Correo Electrónico Seguro para la Red de Investigación Educativa, con las características ya explicadas en este Anexo, se trabajó en dos frentes: configuración del servicio de correo electrónico para acceso a través de un cliente Web y de un Cliente Remoto.

El servicio se implemento con la versión de Openwebmail 2.21-3, sobre un servidor Web Apache 1.3.29.0.2-4, con OpenSSL 0.9.7c-5 y libapache-mod 2.8.16-7 (soporte HTTPS para Apache); UW-IMAP 2001a-debian-6 para el acceso remoto a los mensajes; y Exim 4.30-4, como MTA.

El único inconveniente serio que se tuvo en la implementación de esta parte del servicio, fue la necesidad de recurrir a una versión de UW-IMAP (2001a-debian-6), que permitiera tanto conexiones inseguras (a través del puerto 143), como seguras (a través del puerto 993), pues Openwebmail adolecía de soporte para conexiones con SSL; no obstante, dado que ambas aplicaciones residían en la

misma máquina, no había ningún problema en que los procesos que corrieran se comunicaran en el interior del servidor de forma insegura. No había ningún problema con las conexiones seguras de los clientes a través de los navegadores, porque el soporte SSL en éstos, corría por cuenta de Apache.

La otra parte del servicio de Correo Electrónico, tuvo otros inconvenientes, la mayoría de ellos debidos a la necesidad de soportar múltiples agentes de correo de usuario diferentes y a menudo incompatibles entre sí.

La primera versión de Exim instalada, fue la 3.35-1woody2 (stable). Dado que el problema que se tenía, era que se quería permitir que los usuarios remotos pudieran enviar correo desde sus programas cliente, pero no se quería permitir habilitar esta opción de forma insegura (por ejemplo, permitiendo el reenvío abierto desde el servidor), lo que se hizo fue instalar y configurar SMTP STARTTLS/AUTH para permitir el envío controlado.

Con esto en mente, para configurar Exim con soporte TLS, sólo se debía instalar un paquete adicional: exim-tls. La versión que a la fecha instala el sistema es exim-tls 3.35-3woody1 (stable).

Para adicionar esta funcionalidad, lo que se debe hacer es generar un certificado y una llave privada para permitir la creación de un capa segura a través de la cual se intercambie la información de autenticación, hacer esta llave únicamente leible por Exim, indicarle en el archivo de configuración la ubicación de la misma, exigirle a cualquier máquina que solicite envío que se autentique primero, avisar sobre la disponibilidad de TLS a cualquier equipo que haga una solicitud de envío de correo, y finalmente proporcionar un mecanismo de autenticación.

Con esta configuración se le permite a cualquier MUA que soporte tanto TLS (o SSL) como uno de los tipos de autenticación disponibles, el reenvío de correo, una vez fuera proporcionado el nombre de usuario y la contraseña de usuario.

La idea inicial, era que los usuarios no tuviesen que recurrir a una nueva contraseña para autenticarse y enviar correos a través del servidor, sino que se usara la misma que el usuario empleaba para acceder a su cuenta a través del cliente Web de correo.

Con esto en mente, inicialmente se implementó como mecanismo de autenticación CRAM-MD5, pero se tuvo dificultades a la hora de confrontar la contraseña resumida enviada por el cliente, con la que se recuperaba del archivo con los datos de los usuarios del servicio de correo.

Este archivo era generado a partir del /etc/shadow, por lo que se hacía necesario desencriptar (unshadow) las contraseñas en el mismo, para aplicarles luego una función de resumen (hash), y poder hacer la comparación de la contraseña generada con la enviada por el cliente.

Con el esquema anterior, se realizaron pruebas con el cliente de Correo Kmail 1.5.4 (disponible con el entorno de escritorio KDE 3.1.5), con resultados satisfactorios. No obstante, dado que el MUA que existe por defecto en más del 90% de los computadores de escritorio del mundo es Microsoft Outlook Express, se hicieron pruebas con la versión 6.0 de este programa, y el mecanismo de autenticación falló.

Outlook Express 6.0 es el cliente de correo más popular entre computadores que funcionen con el sistema operativo Windows 98/2000/XP, pues se actualiza junto al Internet Explorer 6.0, que es a su vez el navegador más utilizado del mundo. En esta versión, Outlook adolece de opciones para implementar mecanismos de autenticación como CRAM-MD5 o DIGEST-MD5, aunque permite dos opciones en su configuración: iniciar sesión con o sin contraseña de autenticación segura.

Con la intención de que el mecanismo PLAIN pudiese funcionar con la última opción (sin contraseña de autenticación segura), se implementó este mecanismo, que tenía como principal desventaja el hecho de que ahora las contraseñas se debían almacenar en un archivo en texto plano en el servidor. De nuevo, el sistema no funcionó.

En la búsqueda del origen de este problema y su eventual solución, se encontró en múltiples listas de discusiones, que la raíz del problema se hallaba en que MS Outlook sólo era compatible con los mecanismos de autenticación LOGIN y SPA/NTLM (*Secure Password Authentication / NT LAN Manager*, Autenticación de Contraseña Segura / Administrador LAN NT) sobre SMTP.

Si bien llegados a este punto, lo ideal hubiese sido emplear servicios de directorio o bases de datos para solucionar el problema, estas alternativas se descartaron por el tiempo de implementación que supondrían, en la medida que habría que entrar a evaluar diferentes alternativas, estudiarlas y probarlas con cierto grado de profundidad.

Se optó por seguir buscando una solución pronta al problema, y se encontró con que, a pesar de tener un desarrollo orientado a estándares, la version 4 de Exim, incluía un soporte para esta forma de autenticación propietaria.

El lanzamiento de Exim 4, supuso un cambio en varios de los elementos que constituían al MTA, el más significativo de los cuales, fue la disgregación del archivo de configuración (/etc/exim/exim.conf) en un conjunto de archivos, que facilitaba la modularización de las tareas de administración del servicio de correo electrónico. Así que se instalo Exim 4.3 (testing) y exim4-daemon-heavy (y no exim4-daemon-light), porque este era el paquete que proporcionaba el soporte para la autenticación SPA.

Para que la autenticación SPA funcione, del lado del cliente, y según pruebas efectuadas con MS Outlook 6.0, se debe tener inactiva la opción “Iniciar sesión usando autenticación de contraseña segura" en la sección de “configuración de cuentas”. En el lado del servidor, el archivo que contiene las contraseñas debe estar en el formato: usuario contraseña (separados por un espacio). Este archivo debe estar en texto plano, y su propietario y grupo, deben ser Exim (Debian-exim,

en nuestro caso). Por razones de seguridad, se recomienda que los permisos sobre este archivo estén ajustados a 400. En realidad lo que importa es que el MTA, tenga permisos de lectura sobre este archivo, bien sea como usuario o como grupo.

Observación: Cada vez que se desee modificar la sección /etc/exim4/conf.d/auth, se debe detener el servicio de SMTP y correr el script: /usr/sbin/update-exim4.conf para hacer efectivos los cambios.

5. RECOMENDACIONES Y TRABAJO FUTURO

A pesar de que el Servicio de Correo Electrónico satisface el conjunto mínimo de requisitos de seguridad con los que debería contar cualquier servidor destinado para este fin: conexiones exclusivamente a través de SSL para los clientes del servicio Web de correo, conexiones IMAP seguras y autenticación SMTP con TLS, para evitar que sea utilizado como fuente de SPAM hacia Internet, y garantizar que la información entre el servidor y el cliente vaya cifrada, esto no es suficiente para tener un servicio lo más seguro posible.

Se recomienda la inclusión de mecanismos que prevengan que a los usuarios del sistema les llegue SPAM, bien sea instalando y configurado una utilidad con tal fin, o empleando los esquemas de suscripción a listas negras que maneja Exim (una opción menos efectiva).

A pesar de que el servicio implementado no tendría problemas en su funcionamiento debido a ataques de virus y gusanos (a menos, claro está, que se trate de una denegación de servicio y se saturen los recursos de la máquina), se debe implementar un servicio de antivirus que opere en el servidor, y que impida que los clientes envíen y reciban mensajes de correo electrónico infectados.