Universidad de Alcalá

Departamento de Automática

ARQUITECTURA DE REDES
Laboratorio

Práctica 6:
"Correo electrónico

Grado en Ingeniería Informática
Curso 2022/23

1. OBJETIVOS

En esta práctica se va a hacer uso de los protocolos de correo electrónico más habituales. En concreto, se van a utilizar:

- SMTP (Simple Mail Transfer Protocol), que es el que usan los servidores para transmitirse mensajes entre sí, así como los agentes de usuario para transmitir al servidor los mensajes que envía el mismo.
- POP3 (Post-Office Protocol), versión 3. Es uno de los protocolos que puede usar un agente de usuario para obtener los mensajes que un usuario recibe en su servidor de correo.
- IMAP (Internet Messaging Application Protocol). Otro de los protocolos de lectura de mensajes recibidos.

Lo que se pretende es que el alumno adquiera soltura con estos protocolos, sea capaz de distinguir las posibilidades que ofrece cada uno y la distinta filosofía de funcionamiento.

2. ACTIVIDADES

- Creación de cuentas en servicios de correo electrónico accesibles al alumno. Esta actividad no será necesaria si el alumno ya dispone de ellas.
- Conexión con los servidores por medio de telnet o de openssl.
- Manejo de la utilidad openssl para recodificar texto entre los formatos base64 y código ASCII normal.
- Envío y recepción de mensajes por medio de SMTP y POP3 e IMAP respectivamente.

3. INTRODUCCIÓN TEÓRICA

A continuación se realiza una breve exposición de las características más sobresalientes de los protocolos SMTP, POP3 e IMAP. El alumno ampliará **necesariamente** esta información recurriendo (al menos) a los documentos de descripción RFC de cada uno de los protocolos. El documento RFC para cada protocolo se encuentra indicado en su apartado.

SMTP

El protocolo SMTP es el protocolo básico que utilizan los servidores de correo para enviarse mensajes entre sí. Está descrito en el documento RFC 5321 y utiliza mensajes en formato ASCII de 7 bits. Esto incumbe tanto a los comandos, como a

las cabeceras y al cuerpo de los mensajes. El protocolo SMTP utiliza el puerto 25. La siguiente tabla lista algunos de los comandos que incluye.

Comando	**Descripción**
HELO	Identificación del cliente, generalmente con un nombre de dominio.
EHLO	Permite al servidor declarar su aceptación de comandos ESMTP (Extended Simple Mail Transfer Protocol).
MAIL FROM	Emisor del mensaje.
RCPT TO	Receptor(es) del mensaje.
TURN	Permite al cliente y al servidor invertir los roles y enviarse mensajes en sentido contrario sin tener que iniciar una nueva conexión.
ATRN	Authenticated TURN. Tiene como parámetros opcionales uno o más dominios. Debe ser rechazado si la sesión no ha sido autenticada.
SIZE	Proporciona un mecanismo por el que el servidor SMTP puede indicar el máximo tamaño de mensaje admitido. Un cliente no debe enviar mensajes de mayor tamaño que el indicado por el servidor.
ETRN	Es una extensión de SMTP. ETRN lo envía un servidor SMTP para solicitar a otro servidor el envío de los mensajes que tenga.
PIPELINING	Permite enviar una sucesión de comandos encadenados sin esperas de respuesta a cada comando.
DATA	Lo envía el cliente para indicar que inicia el envío del contenido del mensaje.
DSN	Comando ESMTP que habilita el envío de notificaciones de estado del envío (Delivery Status Notification).
RSET	Anula la transacción completa y reinicializa el buffer.
VRFY	Verifica que un buzón existe. Por ejemplo, 'VRFY Ted' verifica que el buzón 'Ted' existe en el servidor.
HELP	Devuelve una lista de comandos admitidos por el servidor SMTP.
QUIT	Finaliza la sesión.

Transacción de ejemplo:

```
HELO pepito.es
>250 servidor.es
MAIL FROM: pepe@pepito.es
>250 2.1.5. OK
RCPT TO: usuario@servidor.es
>250 2.1.5. Ok
DATA
>354 Start mail input; endi with <CRLF>.<CRLF>
subject: Asunto
Este mensaje es de prueba.
Probamos el protocolo SMTP.

>250 2.0.0 OK: queued as 6D126A066
QUIT
>221 2.0.0 Bye
```

Las líneas que empiezan por > son las respuestas del servidor. El ejemplo presupone la conexión telnet con el puerto 25 del servidor SMTP. El comando "sub j ect" va incluido en el cuerpo del mensaje y especifica el campo 'asunto' del mensaje.

Seguridad

SMTP (y esto es aplicable también a POP3 y a IMAP) es un protocolo sin seguridad. Dado el riesgo que esto representa, algunos servidores comerciales de SMTP como hotmail de Microsoft o gmail de Google le añaden una capa de seguridad. Esto lo pueden hacer de varias maneras. La primera opción consiste en, una vez iniciada una sesión normal de SMTP por medio de telnet al puerto 25, y antes de cualquier intercambio de información, activar la encriptación por medio de TLS (Transport Layer Security). Para ello, el servidor admite el comando STARTTLS. El problema para el acceso manual (por medio de telnet, sin agente de uausario) es que exige al usuariollevar a cabo la negociación de la encriptación manualmente, lo que es casi imposible. Más cómodo, siempre dentro de la primera opción, resulta hacer uso de una utilidad, 'openssl', que puede encargarse de todo el proceso evitándoselo al usuario. La utilidad 'openssl' proporciona múltiples servicios de encriptación relacionados con SSL (Socket Layer Security). Uno de ellos es sustituir al comando telnet en el uso que se le da en esta práctica, y encargarse de la comunicación con un servidor por medio de un canal encriptado. openssl es capaz de negociar la encriptación del canal, por lo que el usuario se limita a intercambiar comandos y respuestas con el servidor de forma análoga a como lo hacía con telnet. El comando de conexión es:

```
openssl s client -starttls smtp -crlf -connect servidor.com:25
```

La segunda opción es iniciar una sesión de SMTP ya encriptada con SSL desde el principio. En este caso, se trata de usar la utilidad openssl de forma similar a la anterior, pero indicando que el puerto de conexión es el 465, que es el que se usa para esto. El comando en este caso es:

```
openssl s client -crlf -connect servidor.com:465
```

Con cualquiera de las dos opciones, una vez establecida la conexión, es preciso autenticarse. Para eso se necesita utilizar el comando AUTH en una de sus múltiples variantes. Una de ellas es la variante LOGIN, mediante la cual, el servidor solicita, en dos pasos, el nombre de usuario y la contraseña del cliente. Esta información se envía recodificada al formato base64, que es una forma de representar un flujo de bits en bloques de 6 bits. La herramienta openssl es capaz de realizar esta recodificación de la siguiente manera:

```
openssl base64
```

De esta manera la utilidad espera por teclado la cadena a recodificar, y la muestra recodificada por pantalla. Una vez introducida, (Enter marca el fin de la cadena), la combinación de teclas <CTRL+D> pone en marcha la recodificación. Este

procedimiento es muy cómodo para recodificar palabras y contraseñas en un momento. Si es preciso recodificar textos completos, también admite entrada y salida desde un archivo. Evidentemente, esta encriptación no se puede hacer con la misma invocación de openssl con la que se ha creado la conexión. Es preciso ejecutarla en paralelo en otro terminal.

Durante la autenticación, los mensajes del servidor también van recodificados en base64, por lo que para entender lo que responde es preciso utilizar openssl para recodificar a texto ASCII. Si la autenticación es correcta, se puede pasar a enviar comandos al servidor. Estos comandos ya van en texto sin recodificar. La sintaxis es ligeramente diferente en algunos casos, pero en lo fundamental se mantiene, ya que sigue siendo el protocolo SMTP. El siguiente es un ejemplo de comunicación en el que se opta por crear una conexión SMTP SSL desde el principio.

```
openssl s_client -crlf -connect servidor.com:465
>CONNECTED(00000000003)

...establecimiento de la conexión ssl...

>220 mx.google.com ESMTP i8sm55243855eeo.16
ehlo usuario
>250-mx.servidor.com at your service, [193.146.8.29]
>250-SIZE 35882577
>250-8BITMIME
>250-AUTH LOGIN PLAIN XOAUTH XOAUTH2
>250 ENHANCEDSTATUSCODES
auth login
>334 VXNlcm5hbWU6
usuario recodificado a base64
>334 UGFzc3dvcmQ6
contraseña recodificada a base64
>235 2.7.0 Accepted
mail from:<usuario@servidor.com>
>250 2.1.0 OK i8sm55243855eeo.16
rcpt to:<destino@otroservidor.com>
>250 2.1.5 OK i8sm55243855eeo.16
data
>354  Go ahead i8sm55243855eeo.16
subject asunto
Cuerpo del mensaje

>250 2.0.0 OK 1347448708 i8sm55243855eeo.16
```

```
quit
>221 2.0.0 closing connection i8sm55243855eeo.16
>read:errno=0
```

Las respuestas del servidor van marcadas con el carácter '>'. En este caso, se está usando una extensión de SMTP llamada ESMTP (Extended SMTP). Es un superconjunto de SMTP, e incorpora algunos comandos necesarios para tratar las características de seguridad vistas más arriba. En concreto, el comando EHLO es una ampliación de HELO, con la misma sintaxis, que sondea si el servidor admite ESMTP. El servidor responde listando funcionalidades que admite. En este caso, la parte que nos interesa es la linea

```
>250-AUTH LOGIN PLAIN XOAUTH XOAUTH2
```

que indica con AUTH LOGIN que la secuencia de login descrita más arriba es admitida. Por lo demás, los comandos son los mismos usados en la sesión de SMTP, con la diferencia de que los nombres de usuario destino y usuario emisor van entre los caracteres o.

POP3

El protocolo POP está definido en el RFC 1939. Tiene 3 fases de funcionamiento. En la primera, la autenticación, el cliente envía los comandos USER y PASS uno a continuación del otro, y se identifica como usuario autorizado. La segunda fase, la transacción, sirve para que el cliente recupere los mensajes. Opcionalmente, también marcará mensajes para borrar. La última fase, la actualización, tiene lugar cuando el cliente ha terminado la sesión, y en ella se borran los mensajes marcados para borrado. Las comunicaciones se realizan por medio de comandos ASCII. La siguiente tabla lista algunos de los comandos más usados.

Comando	**Descripción**
USER <usario>	Nombre de usuario
PASS <contraseña>	Contraseña
QUIT	Finalizar sesión
STAT	Número de mensajes y tamaño total.
LIST <nº de mensaje>	Número del mensaje y su tamaño. Si no se proporciona número de mensaje, lista todos.
RETR nº de mensaje	Descargar mensaje
DELE nº de mensaje	Borrar mensaje
TOP mensaje líneas	Muestra las primeras "líneas" líneas del mensaje número "mensaje". Incluye la cabecera.
NOOP	No-operación
RSET	Deshace los cambios hechos en la sección, incluido el borrado de mensajes.

Transacción de ejemplo:

```
USER pepe
>+OK Ñame is a valid mailbox
PASS LaContrasenia
>+OK Mailbox locked and ready
LIST
>+OK sean listing follows
>1 23941
>2 2411
>3 16523
>4 892034
>.
QUIT
>+OK
```

De nuevo, las líneas que empiezan por > son las respuestas del servidor. El ejemplo presupone la conexión telnet con el puerto 110 del servidor POP3.

Por cuestiones de seguridad, algunos servidores también utilizan conexiones de tipo SSL. Igual que en el caso de SMTP con SSL, se puede usar la utilidad openssl para establecer una conexión segura con el servidor POP3. El comando en concreto sería:

```
openssl s_client -connect pop.gmail.com:995
```

En este caso, la utilidad openssl se utiliza como cliente seguro (s_client) para conectarse (-connect) al servidor POP3 de Gmail. El puerto para este tipo de conexión es el 995.

En el caso de cuentas en servidores de Microsoft (cuentas de live.com, hotmail.com, outlook.com), los servidores suelen funcionar sobre Windows. En ellos, la secuencia retorno de carro-linea nueva se trata de forma algo diferente a como se trata en sistemas de tipo Unix. Para compensar esto, si el servidor de correo que se use en la práctica es de Microsoft, el comando "openssl" debe incluir una opción que le indique esta circunstancia. En concreto, el comando a usar quedaría así:

```
openssl s_client -crlf -connect pop.gmail.com:995
```

IMAP

El protocolo IMAP es un protocolo que permite a un cliente de correo gestionar los mensajes de un usuario en un servidor. Ha ido evolucionando a lo largo del tiempo, y actualmente se encuentra en su versión 4, revisión 1, aunque alguna versión anterior ha experimentado una evolución en paralelo, como la versión 2 bis. La versión 4 revisión 1 se encuentra descrita en el documento RFC 3501.

El protocolo especifica comandos para que el cliente de correo pueda gestionar los mensajes del usuario. La principal diferencia con POP3 es que no es necesario descargar los mensajes del servidor (aunque es posible hacer algo equivalente, si se precisa). IMAP es un protocolo más complejo que POP3, ya que permite una gestión

más sofisticada del correo. La siguiente tabla lista algunos de los comandos de IMAP4. La descripción es necesariamente demasiado sencilla, ya que generalmente la funcionalidad es muy amplia. Para comprender la funcionalidad completa de cada comando es preciso consultar el documento RFC. Todos los comandos deben empezar por una etiqueta arbitraria. Esto es porque el protocolo permite ejecutar varios en paralelo, de forma que el servidor utiliza las etiquetas cuando responde para indicar a qué comando corresponde la respuesta.

IMAP trabaja con una jerarquía de buzones de correo, y antes de acceder a los mensajes de un buzón es preciso seleccionarlo. Los mensajes de un buzón tienen dos identificadores: su número de secuencia (el orden en el buzón) y su UID (un identificador global único). Para algunos comandos basta el número de secuencia, pero para otros es preciso utilizar el UID.

Comando	**Parámetros**	**Descripción**
LOGIN	usuario contraseña	Entrada al sistema
LIST	referencia nombre de buzón	Muestra el listado de carpetas de correo del usuario
SELECT	nombre de buzón	selecciona un buzón para poder acceder a sus mensajes
EXAMINE	nombre de buzón	Igual que Select, pero no realiza modificaciones
FETCH	número de secuencia elemento(s) del mensaje	Obtiene elementos de un mensaje identificado por su número de secuencia.
SEARCH	criterio de búsqueda	muestra los mensajes que cumplen el criterio de búsqueda
STORE	número de secuencia característica valor de la característica	permite asignar el valor especificado a la característica especificada del mensaje identificado por su número de secuencia.
COPY	número de secuencia buzón	copia (sin borrar) el mensaje identificado por su número de secuencia al buzón indicado
EXPUNGE		borra los mensajes con la característica "\Deleted"

IMAP usa el puerto 143. También se usa IMAP con SSL. En este caso, el puerto es el 993. Para esta conexión, de nuevo es necesario usar el comando openssl.

Transacción de ejemplo:

```
1 login "usuario"
"contraseña" >1 OK LOGIN
completed. 1 list "" "*"

>* LIST (\Marked) (\HasNoChildren) "/"  INBOX
     ...resto de listado de buzones... >1 OK LIST
```

```
completed. 1 select INBOX >*1 EXISTS >*1 RECENT
>*FLAGS (\Seen \Answered \Flagged \Deleted \Draft;
     ...resto de listado del buzón... >1
OK [READ-WRITE] SELECT completed. 1
fetch 1 body >* 1 FETCH (BODY[])
}3094}
     ...cuerpo del mensaje... >1 OK
FETCH completed. 1 store 1
+flags.silent (\Deleted) >1 OK STORE
completed. 1 expunge >* 1 EXPUNGE
>* 0 EXISTS
>1 OK EXPUNGE completed.
1 logout
>* BYE  Mensaje despedida.
>1 OK LOGOUT completed.
```

La secuencia anterior lo primero que hace es un LOGIN en el servidor. Algunos servidores requieren las cadenas usuario y contraseña entre comillas (") y otros no. Una vez en el servidor, el comando LIST solicita los buzones del usuario. La respuesta del servidor es un listado con los buzones del usuario y sus características. A continuación se selecciona el buzón INBOX. El servidor indica, entre otras cosas, el número de mensajes total y el número de mensajes recientes. El comando FETCH se usa para obtener el cuerpo del mensaje, que incluye, no sólo el texto del mensaje, sino también otra información adicional. Con el comando STORE se marca el mensaje 1 (número de secuencia, no UID) para borrar. El borrado no ocurre inmediatamente, sino cuando se emite el comando EXPUNGE. La respuesta del servidor es listar el número de mensajes total, que, como se puede ver, ha pasado de 1 a 0 al borrarse el que se marcó para borrar. El comando LOGOUT termina la transacción.

EJERCICIOS

En los siguientes ejercicios puede ser necesario usar alguna de las siguientes direcciones.

Servicio de correo	**Servidor SMTP**	**Servidor P0P3**	**Servidor IMAP**
Universidad	correo.uah.es	correo.uah.es	correo.uah.es
Gmail	smtp.gmail.com	pop.gmail.com	imap.gmail.com
Hotmail	smtp.live.com	pop3.live.com	-

(-) Hotmail no proporciona servicio IMAP.

Los puertos de conexión se resumen en la siguiente tabla:

Protocolo	Puerto
SMTP	25
SSMTP [SMTP con SSL)	465
P0P3	110
P0P3 con SSL	995
IMAP	143
IMAP con SSL	993

Para realizar una conexión con STARTTLS, el procedimiento es arrancar una conexión en el puerto normal (por ejemplo, para SMTP, el 25) y luego ejecutar el comando STARTTLS. Como se ha dicho más arriba, esto implica que el usuario debe gestionar él mismo la negociación de las claves de encriptación, lo cual es muy complicado. Lo más aconsejable es usar la utilidad openssl con la opción -starttls.

Ejerciciol.

Se trata de utilizar el protocolo SMTP de la forma descrita más arriba para enviar mensajes de correo. Estos mensajes serán leídos en los ejercicios siguientes. Será necesario disponer de una cuenta de correo en algún servidor. Se puede usar la cuenta del alumno en el servidor de la Universidad, o también cuentas en algún servicio de correo disponible, como hotmail o gmail. Hay que tener en cuenta que en ocasiones se deberá teclear la contraseña desprotegida en un terminal, por ejemplo para recodificarla a base64, por lo que, si se desea evitar esto se recomienda realizar la recodificación previamente en un lugar discreto (en casa, por ejemplo).

Ejercicio 2.

Ahora se trata de comprobar la funcionalidad del protocolo P0P3. Para ello, se obtendrá alguno de los mensajes enviados. Será preciso conectarse al servidor, autenticarse, obtener el número de mensajes, analizar alguno de sus componetes con el comando TOP, descargarlo del servidor y borrarlo. Puede ser útil en caso de errores utilizar el comando RSET.

Ejercicio 3.

De forma análoga al ejercicio 2, comprobar la funcionalidad de los comandos del protocolo IMAP. Conectarse al servidor, autenticarse, y analizar los buzones del usuario. El mensaje enviado anteriormente seguramente haya sido colocado en el buzón INBOX. Situarse en él, localizar el mensaje (tal vez sea necesario usar el comando SEARCH), obtener sus datos (cabecera, cuerpo), marcarlo para borrar y borrarlo.