Práctica 6

Alumno 1: Oganesyan Aram
Alumno 2: Muñoz García, José María

Titulación: Grado de Ingeniería de Computadores

PC de la práctica: Mi PC

Paso 1: Tráfico correo simple (p6smtp1.pcapng):

Ejercicio 1. Indique los puertos usados por el cliente y el servidor para la comunicación.

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
24	10.729806	192.168.43.135	192.168.43.173	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>
25	10.729921	192.168.43.173	192.168.43.135	NBNS	253	Name query response NBSTAT
26	11.889604	192.168.43.135	192.168.43.173	SMTP	92	S: 220 Aram-ASUS ESMTP SubEthaSMTP null
27	11.890132	192.168.43.173	192.168.43.135	SMTP	64	C: EHLO RAS
28	11.896298	192.168.43.135	192.168.43.173	SMTP	107	S: 250 Aram-ASUS | 250 8BITMIME | 250 AUTH LOGIN | 250 Ok
29	11.897257	192.168.43.173	192.168.43.135	SMTP	87	C: MAIL FROM: <aram@practicas.com>
30	11.904686	192.168.43.135	192.168.43.173	SMTP	62	S: 250 Ok
31	11.905304	192.168.43.173	192.168.43.135	SMTP	83	C: RCPT TO: <profesor@rysd.es>

```
> Frame 29: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
> Ethernet II, Src: LiteonTe_3b:59:9c (20:68:9d:3b:59:9c), Dst: Azurewav_ff:46:df (40:e2:30:ff:46:df)
> Internet Protocol Version 4, Src: 192.168.43.173, Dst: 192.168.43.135
v Transmission Control Protocol, Src Port: 50709, Dst Port: 25, Seq: 11, Ack: 92, Len: 33
    Source Port: 50709
    Destination Port: 25
    [Stream index: 6]
    [TCP Segment Len: 33]
    Sequence number: 11    (relative sequence number)
    [Next sequence number: 44    (relative sequence number)]
    Acknowledgment number: 92    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 68
    [Calculated window size: 17408]
    [Window size scaling factor: 256]
    Checksum: 0xa496 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (33 bytes)
> Simple Mail Transfer Protocol
```

Ejercicio 2. Use la opción **Follow TCP Stream** de Wireshark para observar el diálogo completo que han mantenido el cliente de correo y el servidor. Adjunte una captura de pantalla donde se observe dicho diálogo.

Wireshark · Follow TCP Stream (tcp.stream eq 6) · Captura Cliente 1.pcapng

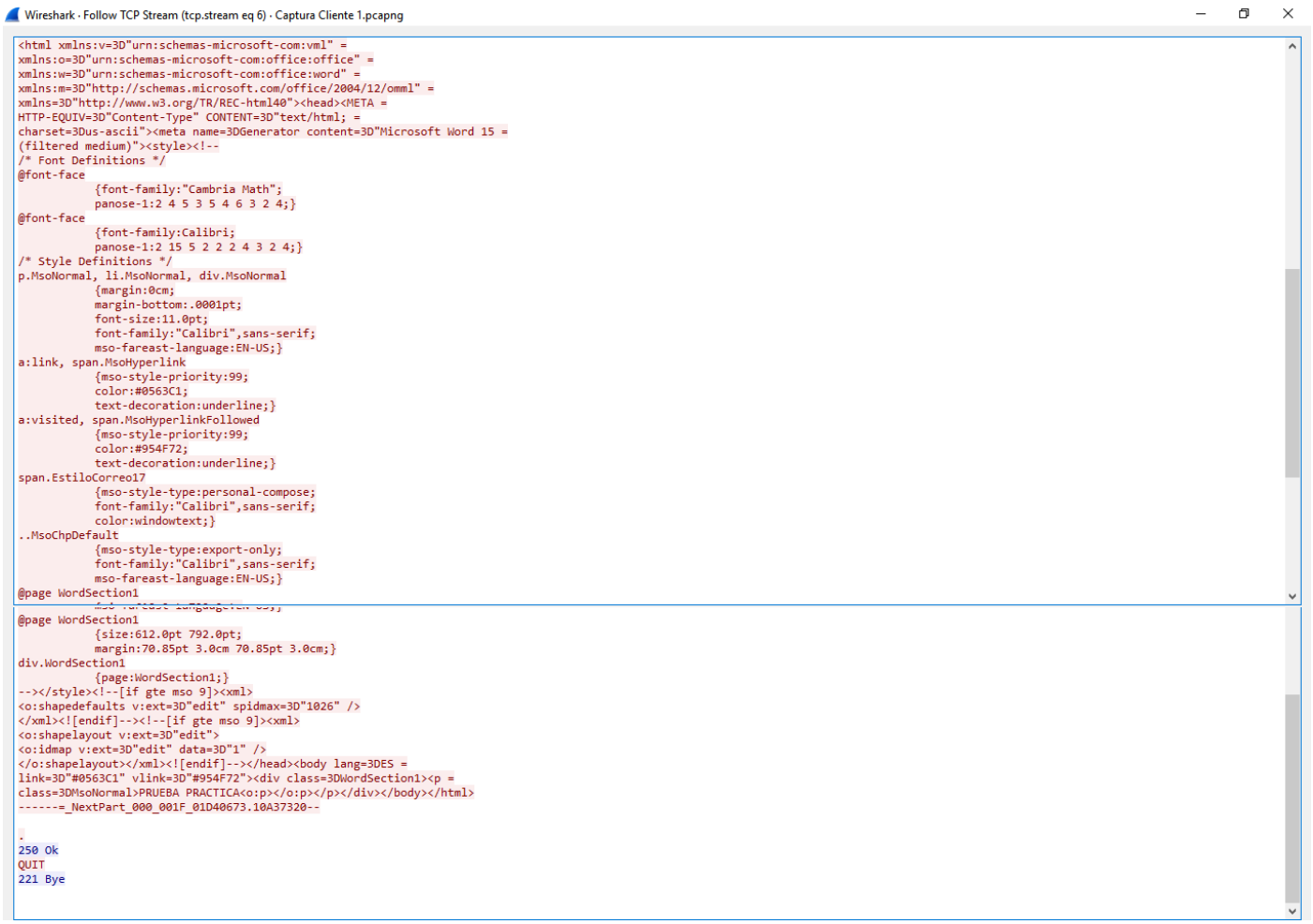
```
220 Aram-ASUS ESMTP SubEthaSMTP null
EHLO RAS
250-Aram-ASUS
250-8BITMIME
250-AUTH LOGIN
250 Ok
MAIL FROM: <aram@practicas.com>
250 Ok
RCPT TO: <profesor@rysd.es>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "Rafa" <aram@practicas.com>
To: <profesor@rysd.es>
Subject: PONTE YA EL SERVIDOR
Date: Sun, 17 Jun 2018 19:40:34 +0200
Message-ID: <001e01d40662$4d1a2df0$e74e89d0$@practicas.com>
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_NextPart_000_001F_01D40673.10A37320"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdQGYjqX1UonIcb+Sye9Rd1OCHCZyA==
Content-Language: es

This is a multipart message in MIME format.

------=_NextPart_000_001F_01D40673.10A37320
Content-Type: text/plain;
        charset="us-ascii"
Content-Transfer-Encoding: 7bit

PRUEBA PRACTICA


------=_NextPart_000_001F_01D40673.10A37320
Content-Type: text/html;
        charset="us-ascii"
Content-Transfer-Encoding: quoted-printable
```



Ejercicio 3. Explique cada una de las instrucciones enviadas por el cliente, indicando para qué se usa. ¿Cómo determina el servidor cuándo termina el cuerpo del correo?

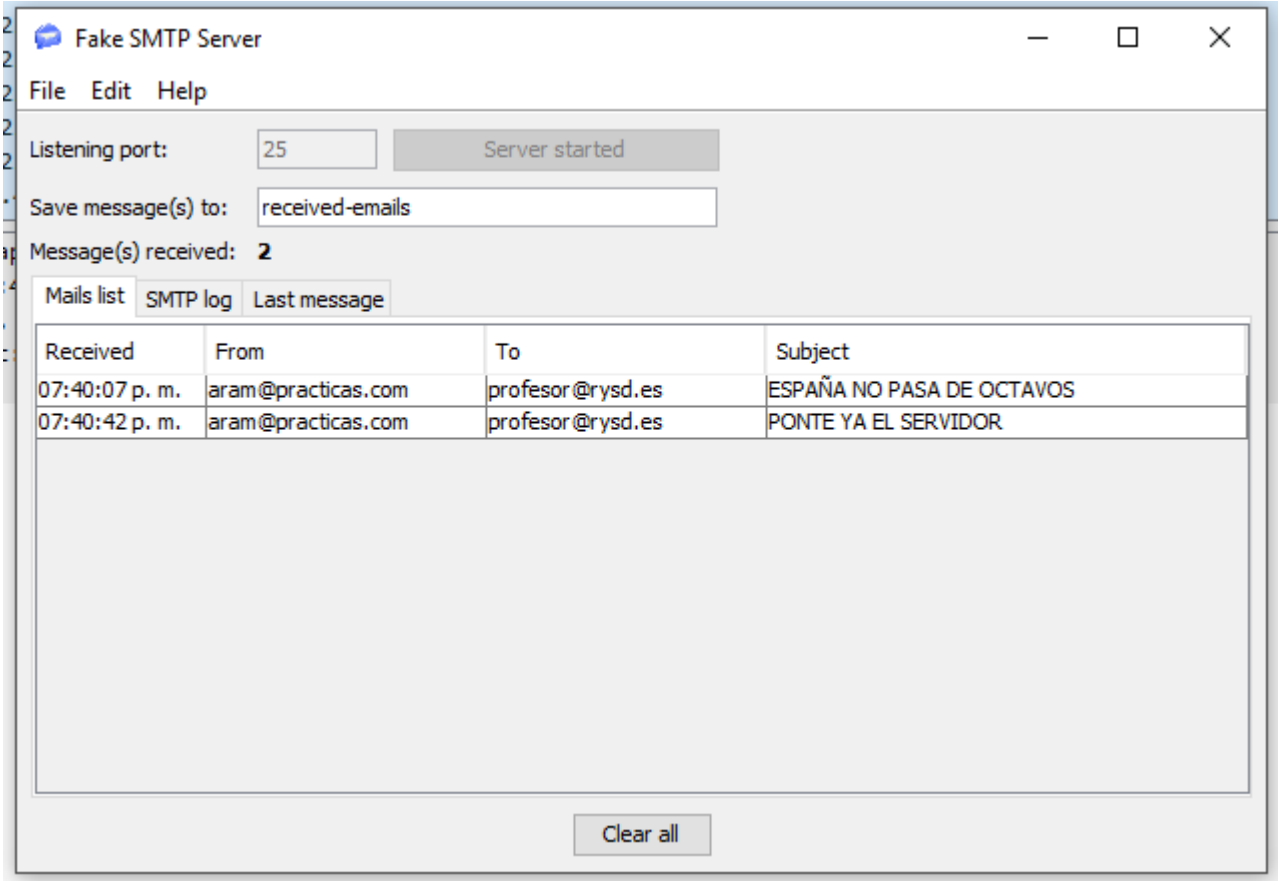
Primero envía un ‘’HELO’’ con el nombre del equipo al que quiere conectarse. Una vez que realiza la conexión, empieza la autenticación estableciendo una llave de cifrado común.
Después se envían mensajes de autenticación de cifrados entre ellos para comprobar que usan el mismo cifrado.
Luego el cliente empieza a enviar datos y recibir confirmaciones. Primero el email, luego el receptor y al final el cuerpo o los datos.
Los siguientes son el DATA o mensaje entero con todos sus parámetros y los que contienen.
Una vez confirmada la llegada envía una señal de finalización y el servidor cierra la conexión.
Y finalmente determina que ha terminado cuando encuentra un <CR><LF>. <CR><LF>

Ejercicio 4. ¿Qué significado tiene el código que aparece al principio de cada respuesta? (el primer dígito) ¿Qué valores puede tener? (para contestar a esto puede consultar cualquier documentación).

Los códigos los envía el servidor. Según el valor cambiará su significado, si estos números empiezan por 4 o 5 sabemos que se ha tratado de algún tipo de error. Si empieza por 2, 0, 3 sabremos que es un código normal, que indica un buen funcionamiento. (Como podemos ver en las diapositivas).
En nuestra captura aparecen los siguientes códigos:

- **220: El servidor SMTP nos ha reconocido y está listo.**
- **250: Solicitud aceptada.**
- **354: Acepta el resto de datos que enviemos como texto libre.**
- **221: La solicitud de cierre de la comunicación ha sido aceptada.**

Ejercicio 5. Localice la cabecera del mensaje y explique qué información contiene cada elemento de dicha cabecera. ¿Por qué se repite el destinatario en la cabecera del mensaje?



La cabecera contiene los siguientes contenidos:

- **El emisor es aram@practicas.com**
- **El receptor profesor@rysd.es**
- **El asunto PONTE YA EL SERVIDOR.**
- **El otro asunto es de otra prueba que no salió bien, y llegó en el segundo intento.**

Además de estos datos también podemos encontrar información sobre la fecha y otros datos.

Se repite dos veces porque forma parte de todo el mensaje y en la segunda forma parte de un comando de identificación del usuario.

Paso 2: Tráfico correo avanzado (p6smtp2.pcapng):

Ejercicio 6. Observe la cabecera del mensaje. ¿Por qué no aparece el destinatario oculto en dicha cabecera? ¿Dónde aparece el destinatario oculto?

Wireshark · Follow TCP Stream (tcp.stream eq 3) · Captura Cliente 2.pcapng

```
220 Aram-ASUS ESMTP SubEthaSMTP null
EHLO RAS
250-Aram-ASUS
250-8BITMIME
250-AUTH LOGIN
250 Ok
MAIL FROM: <aram@practicas.com>
250 Ok
RCPT TO: <administrador@rysd.es>
250 Ok
RCPT TO: <alumno@rysd.es>
250 Ok
RCPT TO: <profesor@rysd.es>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "Rafa" <aram@practicas.com>
To: <profesor@rysd.es>
Cc: <alumno@rysd.es>
Subject: SEGUNDA TRAMA PRUEBA
Date: Sun, 17 Jun 2018 19:52:27 +0200
Message-ID: <002901d40663$f5dd3170$e1979450$@practicas.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;
            boundary="----=_NextPart_000_002A_01D40674.B96676A0"
X-Mailer: Microsoft Outlook 15.0
Thread-Index: AdQGY/FemWifsKzVQjWF5QldDC42eA==
Content-Language: es
```

Vemos que el destinatario oculto (CCO) no aparece en la captura, pero si nos fijamos arriba, en las instrucciones RCPT TO sí podemos verlo.

Ejercicio 7. ¿Qué parte del mensaje es la imagen? Haga una captura de pantalla y márquela. ¿Qué codificación se está usando para la imagen? ¿Cuánto ocupará la imagen en el correo con respecto al tamaño original (en porcentaje)? (para contestar a esto puede consultar cualquier documentación).

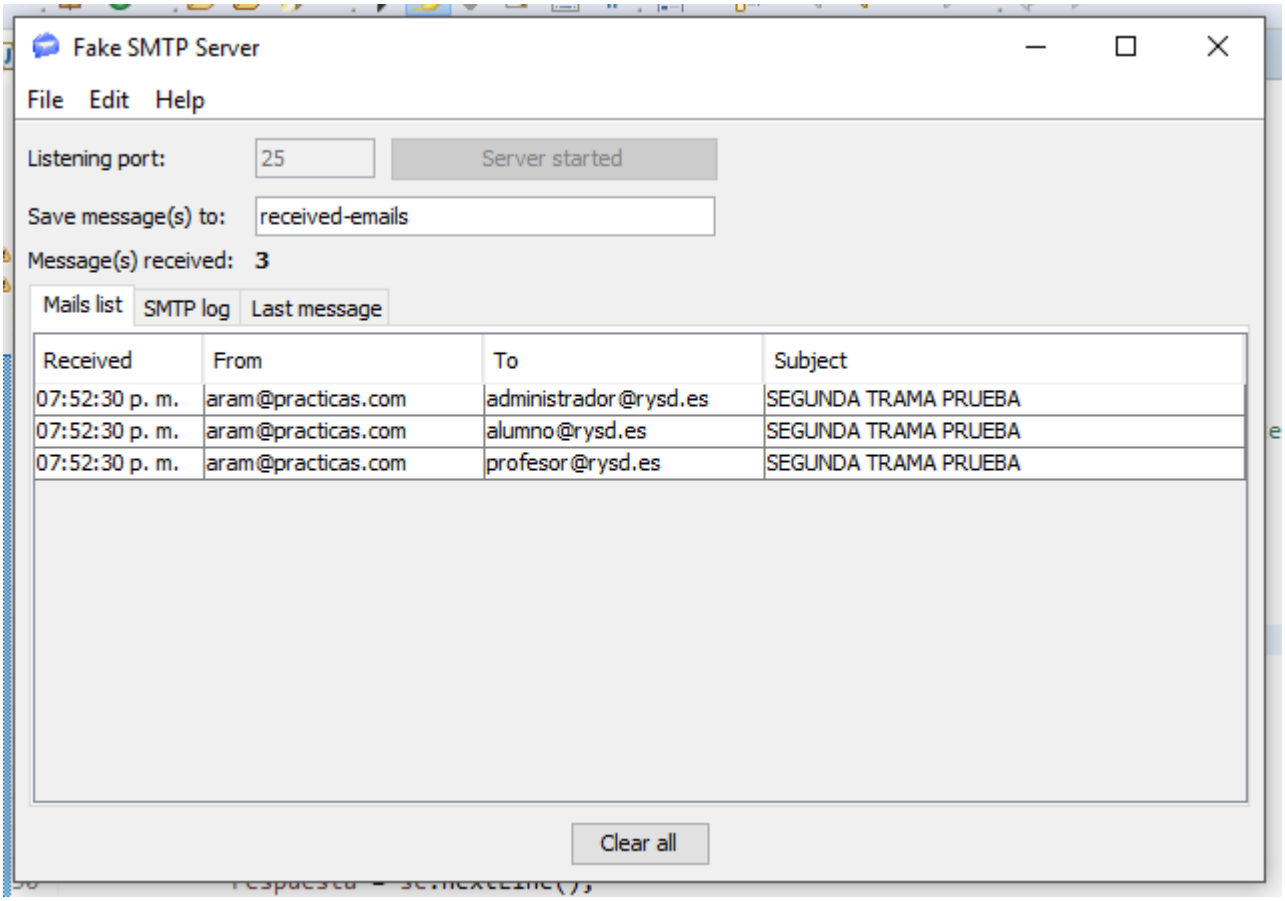
```
------=_NextPart_000_002A_01D40674.B96676A0
Content-Type: image/jpeg;
            name="Practica6.jpg"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
            filename="Practica6.jpg"
```


```
/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAkGBggGBQkIBwgKCQkKDRYODQwMDRoTFBsWHxwhIB8o
JSojJzIuLCQuJR4kNzc2Oiw0LDc4ISs9NzwsNSwxLCwBCQoKBQUFDQUFDSkYEhgpKSkpKSkpKSkp
KSkpKSkpKSkpKSkpKSkpKSkpKSkpKSkpKSkpKSkpKSkpKSkpKSkpKf/AABEIAC8ALwMBIgACEQED
EQH/xAAaAAADAQADAAAAAAAAAAAAAAAADBAUCAAEH/8QAMRAAAQMCAgcHAwUAAAAAAAAAAQIDBAAR
BSESFBUxVJLRBhZBUVNxohOBsSJikaHh/8QAFAEBAAAAAAAAAAAAAAAAAAAAP/EABQRAQAAAAAA
AAAAAAAAAAAAAAD/2gAMAwEAAhEDEQA/APX38eisPKbCXXSnIltII/NY7xx/Qk8g6112aA2ao2zL
hufsKrUErvHH9CTyDrXO8cf0JPIOtUJEluIyXXlaKBle16LvGVBK7xx/Qk8g60aLjUaU99Iaba/A
OC16fqF2oACY6gP1XUL/AMUBuzriG8LOmpKbuHebeAouK4pqbCDHUhbilbr3yG+lsDiMSsLH120u
aLirX9hVFGGQ2ySiOgEgi9vPKgRxpZmxYrTOesHTHsB/tOYXJD+HMEqGloWIvnllUzCm3Nrhh3MQ
0qCT7noa0xgKlypBdUtpAVZkpPgTQWTJZAJLrYt+4VC7RSGpMeOtlYWnSULj7VUTg0FKAnV0mwtc
76k49DahRI7bAISVqJub55UDPZ+SyzBW064htaXDdKjY+FVNcj8Q1zihP4TDkulx1gFZ3kEj8UPY
WH8P81daA4fhpcU4HWAtVtJWkLmta5H4hrnFLbCw/h/mrrXNhYfw/wA1daBnXI/ENc4qL2iebkm0
2wsOrGkSEG/lVHYWH8P81daLGw2LDXpsMhKvO5J/ug//2Q==
```


```
------=_NextPart_000_002A_01D40674.B96676A0--
```

La codificación que se usa es base64, como podemos ver en la captura y la parte del mensaje que corresponde a la imagen es el cuadro marcado en la captura.

Ejercicio 8. Si observa el interfaz gráfico de FakeSMTP, verá que este correo lo ha recibido 3 veces. ¿Por qué?



Si podemos ver que se ha enviado 3 veces, ya que cada mensaje tiene un receptor distinto, que son:

- **administrador@rysd.es**
- **alumno@rysd.es**
- **profesor@rysd.es**

Pero como la dirección y los puertos eran los mismos, entonces han sido captados por nuestro propio servidor de correo.

Paso 3: Tráfico correo correcto generado por código (p6smtp3.pcapng):

Ejercicio 9. Indique qué instrucción del código (`socket`, `connect`, `write`, `read`, `close`) es responsable de generar o tratar cada una de los mensajes generados en la traza de Wireshark. Como envíos y recepciones hay varios, elija un único ejemplo.

Ejercicio 10. ¿En cuales mensajes se usa *piggybacking*? ¿Por qué? En los que no usen esa estrategia, los mensajes de datos ¿confirman algo? ¿El qué?

Paso 4: Tráfico correo incorrecto generado por código (p6smtp4.pcapng):

Ejercicio 11. Cuando el cliente se cierra de forma incorrecta, ¿se intercambian algún mensaje? ¿Cuáles?

Paso 5: Servidor SMTP (sin traza):

Ejercicio 12. Sabría indicar (quizás mediante un uso “inteligente” del cliente desarrollado), si el servidor FakeSMTP es iterativo o concurrente. Justifique la respuesta y añada capturas de pantalla para apoyar su contestación.