GOBIERNO DE ESPAÑA
VICEPRESIDENCIA TERCERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Seminario Web: Uso básico de Wireshark

Ejercicio práctico

017 TU AYUDA EN CIBERSEGURIDAD

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

ÍNDICE

1. Análisis SMTP 3
2. Análisis HTTP 7

ÍNDICE DE FIGURAS

Figura 1 Intercambio de Comandos SMTP 3
Figura 2 Análisis de SMTP desde Wireshark 4
Figura 3 SMTP en profundidad 5
Figura 4 Versión del cliente de correo utilizado 6
Figura 5 Petición HTTP 7
Figura 6 Mensaje HTTP 8
Figura 7 Análisis HTTP desde Wireshark 8
Figura 8 Filtro HTTP 9
Figura 9 Ampliación sobre la pila de protocolos 9

1. ANÁLISIS SMTP

El protocolo SMTP (Simple Mail Transfer Protocol), es un protocolo muy utilizado y asociado a un RFC que data de 1982. Este protocolo se usa para el envío de correo electrónico. La mayoría de los ataques maliciosos, están relacionados con los ficheros adjuntos que son enviados o los enlaces embebidos en el cuerpo del mensaje.

Como se puede apreciar en la siguiente imagen, el protocolo SMTP es un protocolo orientado a línea donde se intercambian mensajes entre el servidor (en rojo) y el cliente (en azul).

El protocolo orientado a línea es un protocolo donde se utiliza un retorno de carro y una línea nueva por cada comando intercambiado.

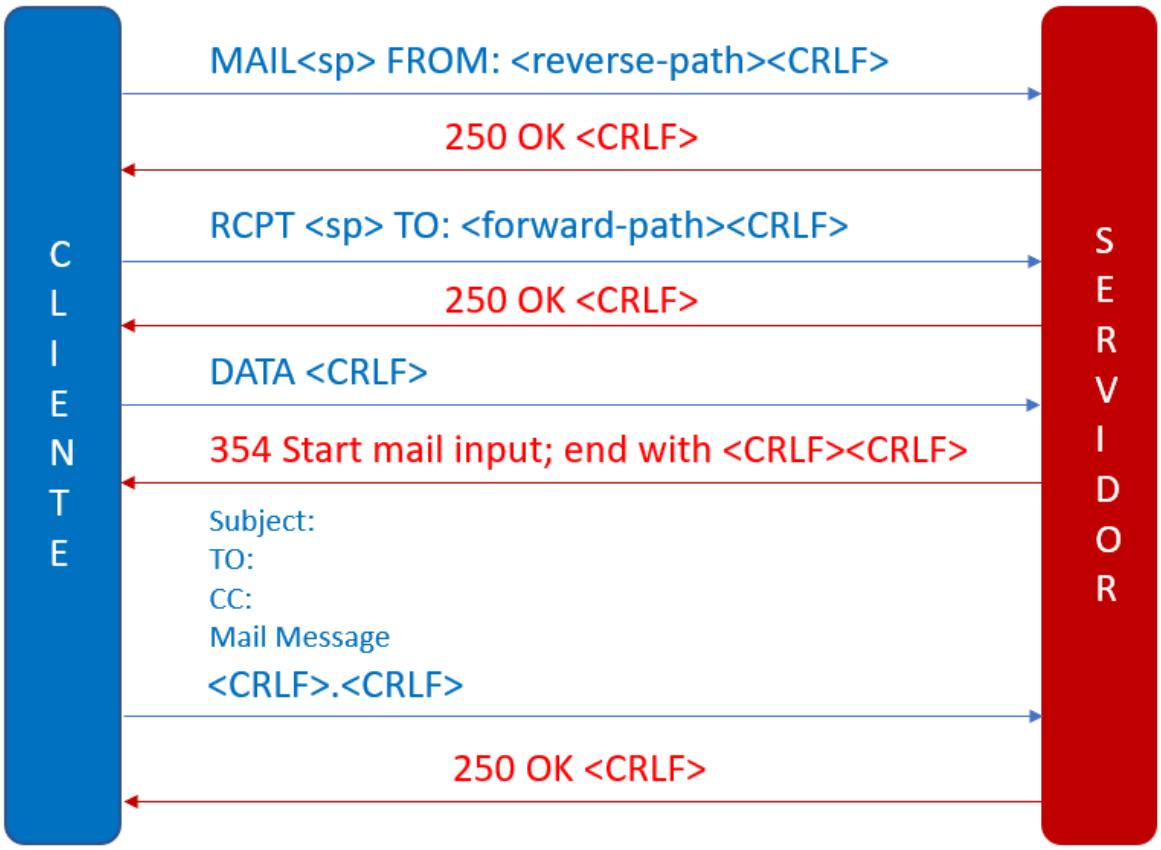

Figura 1 Intercambio de Comandos SMTP

Una transacción SMTP empieza con el comando MAIL desde el cliente hasta el servidor, junto con el parámetro FROM donde se indica a donde se deben enviar los posibles fallos en el envío del correo. SMTP finaliza con un retorno de carro y salto de línea (CRLF). Si el comando MAIL se acepta por parte del servidor, el servidor responderá con el código «250 OK», y esto significa que esta todo correcto.

El siguiente paso es enviar el comando RCPT para identificar una o varias direcciones de correo de destinatario. Si el servidor SMTP acepta esto, responderá «250 OK», en caso contrario responderá con un «550 No such user here», indicando que no ha podido encontrar al destinatario.

El cliente de correo envía el comando DATA para indicar que el siguiente comando será el mensaje. El servidor responde con un «354 Start mail input, end with <CRLF>.<CRLF>», donde el servidor indica al cliente como debe terminar el mensaje, en nuestro caso una línea que contenga «.».Si todo es correcto el servidor devolverá un 250 OK.

Analicemos una captura que contiene una comunicación SMTP:

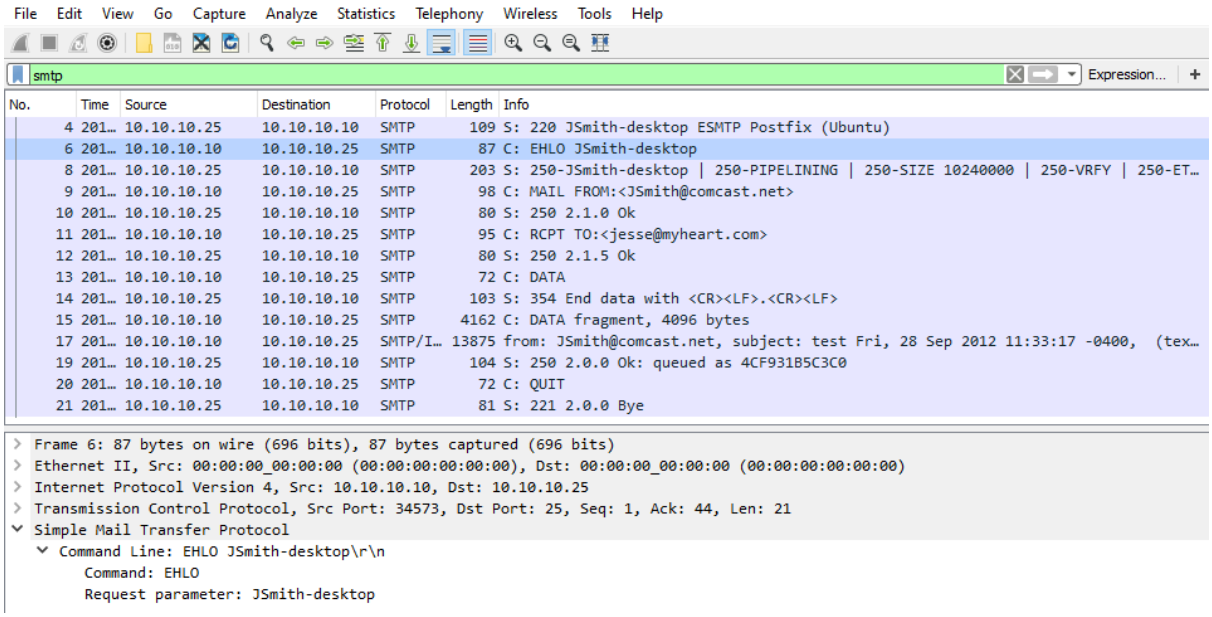

Figura 2 Análisis de SMTP desde Wireshark

Como vemos en la anterior captura, la comunicación empieza previamente con un nuevo comando HELO o EHLO.

En la actualidad ambos se utilizan, pero al utilizar EHLO el servidor responde con características adicionales como PIPELINING, SIZE, HELP and ENHACEDSTATUSCODES como se puede identificar en la siguiente captura:

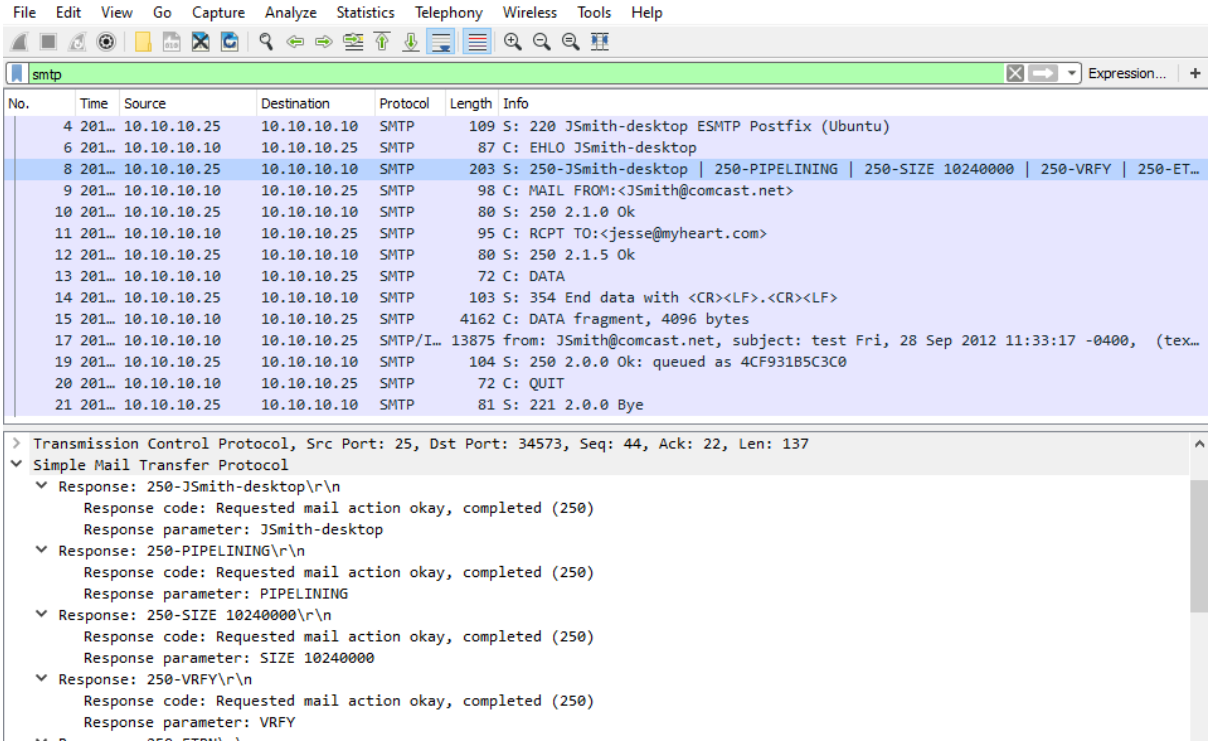

Figura 3 SMTP en profundidad

La sesión comienza con un *three-way handshake* TCP hacia el servidor SMTP, y el servidor responde con un 220 como se puede apreciar en el paquete número 4. El cliente continúa con el mensaje, en el paquete número 6 con un EHLO identificándose como JSmith-desktop y el servidor responde con un «250 OK».

Una vez explicado cómo funciona SMTP te proponemos el siguiente ejercicio, dado la captura smtp_muestra.pcap ¿Sabías decirnos cual es cliente de correo electrónico que supuestamente fue utilizado para enviar el correo electrónico en cuestión?

Sabiendo que el cliente tiene que enviar el mensaje DATA con la información del correo electrónico, incluido las cabeceras, se podría obtener dicha información:

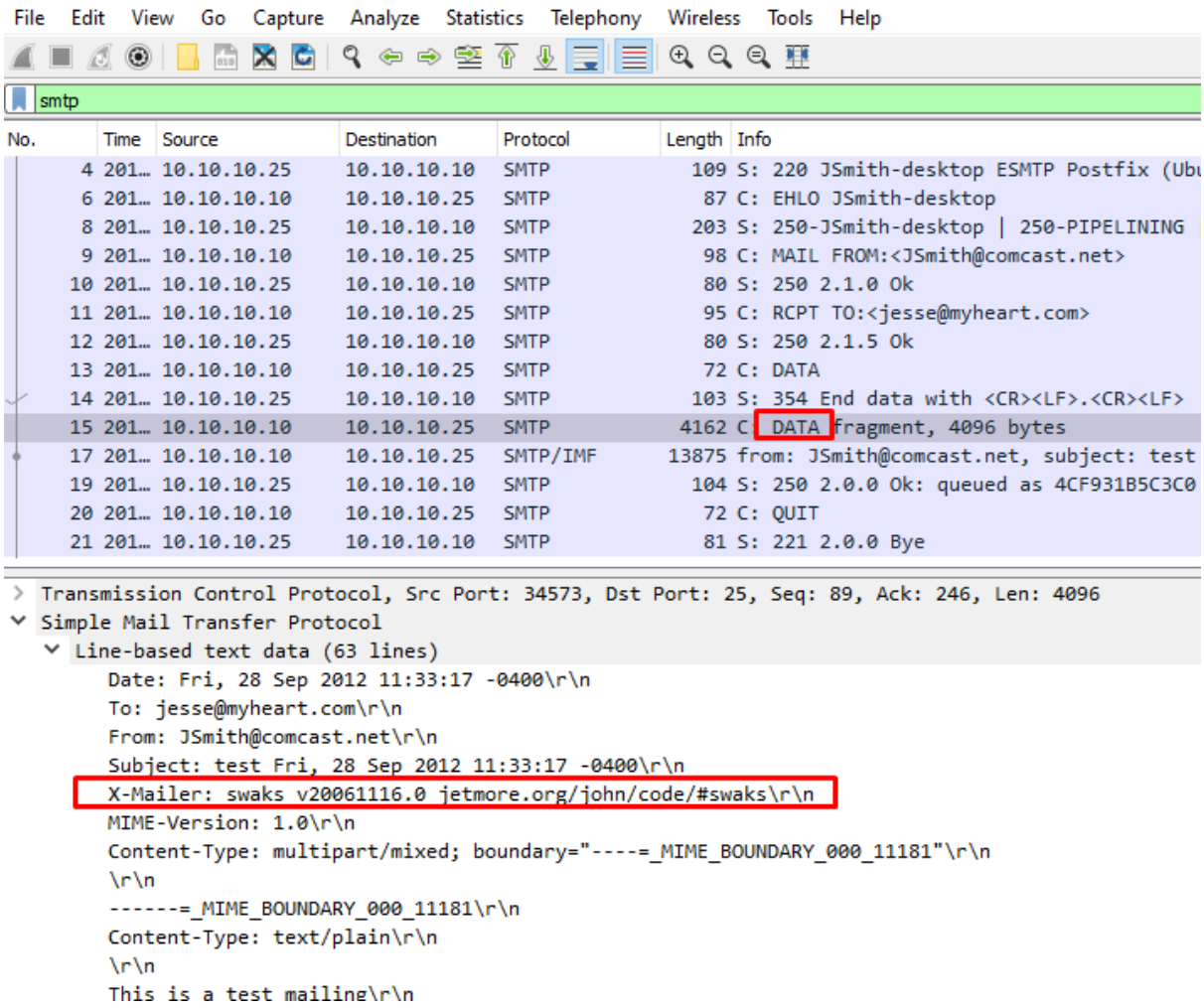

Figura 4 Versión del cliente de correo utilizado

La respuesta sería: «X-Mailer: swaks v20061116.0 jetmore.org/john/code/#swaks»

2. ANÁLISIS HTTP

«Hypertext Transfer Protocol» (HTTP) es un protocolo mundialmente conocido donde inicialmente los ataques fueron perpetrados contra los servidores, pero hoy en día a menudo el objetivo es el navegador para comprometer el host del equipo.

Como veras, HTTP es un protocolo de formato simple, pero el formato de cuerpo de una petición o respuesta HTTP se puede complicar. Los servidores y navegadores son susceptibles de muchas vulnerabilidades y tipos de ataque. HTTP es un protocolo sin estado por que el servidor no mantiene el estado entre las transacciones realizadas en una sesión.

Una petición HTTP comienza con una «Start Line» o línea de empiece que incluye un método, una URL, la versión HTTP y termina con un retorno de carro y nueva línea (CRLF). Existen diferentes métodos HTTP donde el más usado es GET.

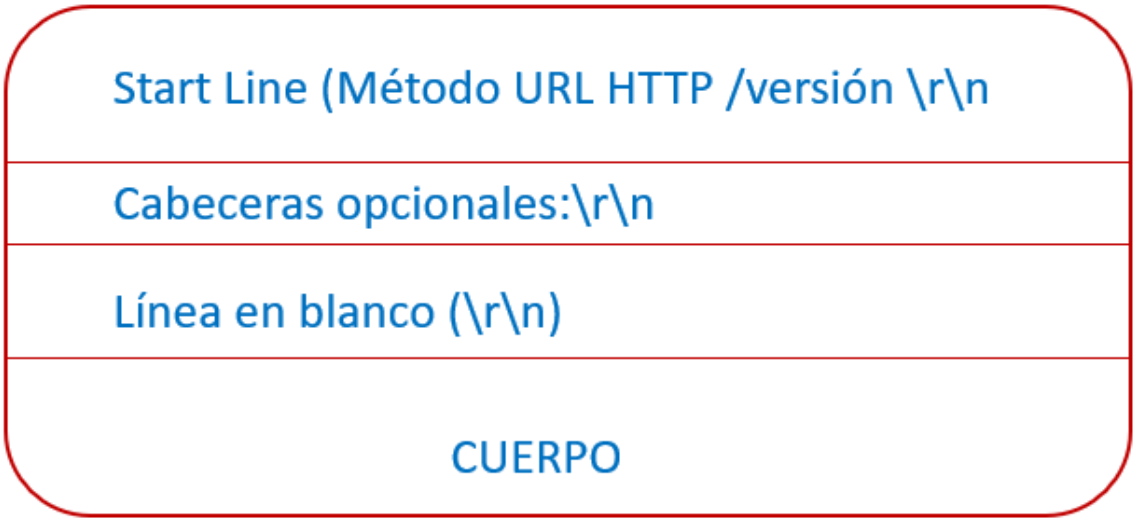

Figura 5 Petición HTTP

Un método GET realiza peticiones de algún tipo de recurso o documento identificado por la URL del servidor. Otro tipo de método es POST, el cual envía datos hacia un servidor especificado por una URL. Puede haber cabeceras opcionales desde que contenido se acepta, idiomas o codificaciones y algunas también por seguridad. Después de las cabeceras podremos encontrar una línea den blanco con un CRLF y finalmente el cuerpo del mensaje.

HTTP y SMTP son protocolos orientados a línea, esto significa que el protocolo utiliza una nueva línea para delimitar los distintos elementos que conforman la petición.

El formato de la respuesta HTTP no es muy diferente al de petición. La única diferencia notable es que la línea de comienzo es la versión, código de estado y la razón-motivo. El campo versión nos indica la versión soportada por el servidor HTTP. El código de estado son 3 dígitos que indican el resultado de la petición. El primer digito indica la clase de códigos tales como éxito o error. El campo razón-motivo indica de manera explícita lo que indica el código de estado. Finalmente, la línea de empiece debe finalizar con CRLF.

Las cabeceras son opcionales de nuevo, pero la mayoría de los servidores las incluyen. Estas deben ser seguidas por una línea en blanco y el cuerpo que es opcional.

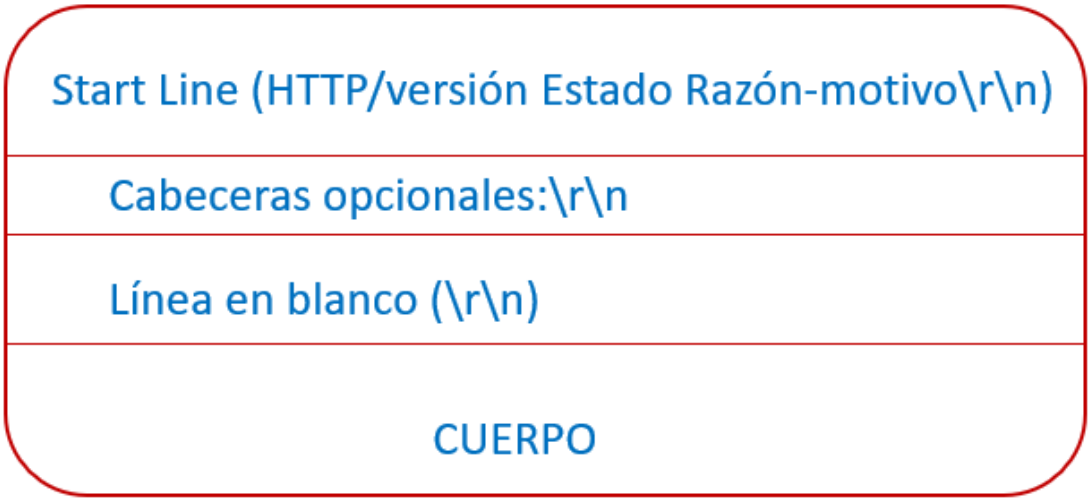

Figura 6 Mensaje HTTP

Dando el fichero http.cap ¿podrías indicarnos que cual sería el servidor involucrado en la petición, es decir, el tipo de servidor utilizado?

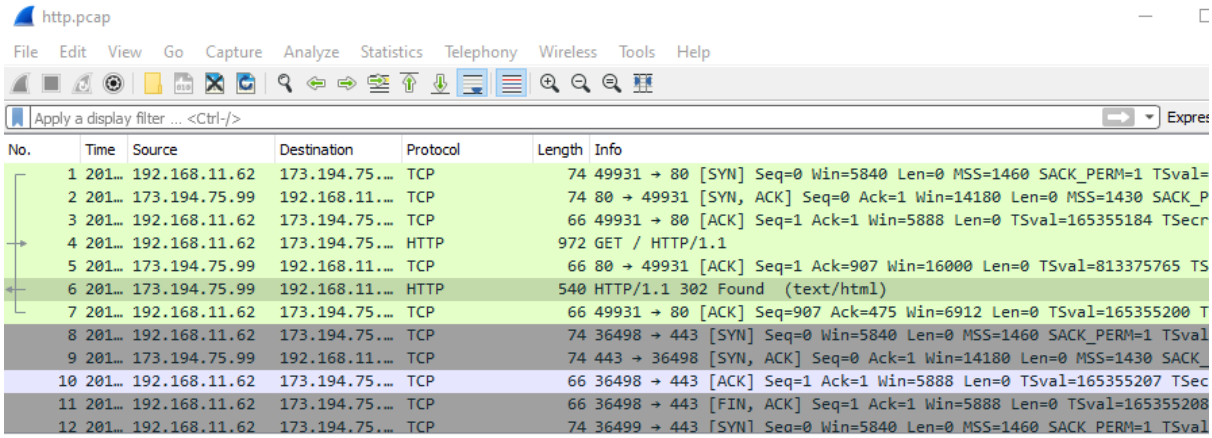

Figura 7 Análisis HTTP desde Wireshark

El primer paso sería filtrar por el protocolo HTTP:

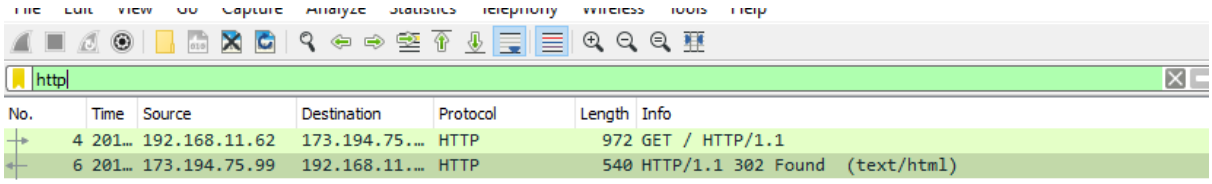

Figura 8 Filtro HTTP

Una vez filtrado nos quedarían dos paquetes HTTP donde deberemos analizar la respuesta del servidor. Abrimos el paquete número 6 podemos ver las cabeceras de la respuesta:

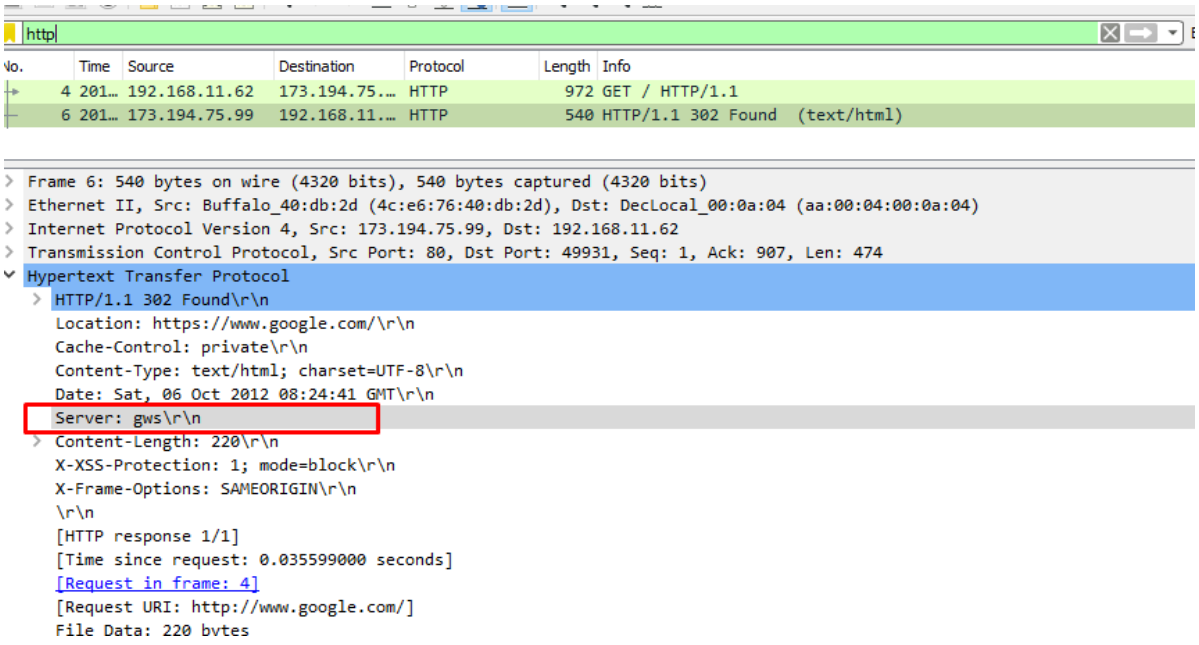

Figura 9 Ampliación sobre la pila de protocolos

Finalmente podemos ver en las caberas el servidor «gws».