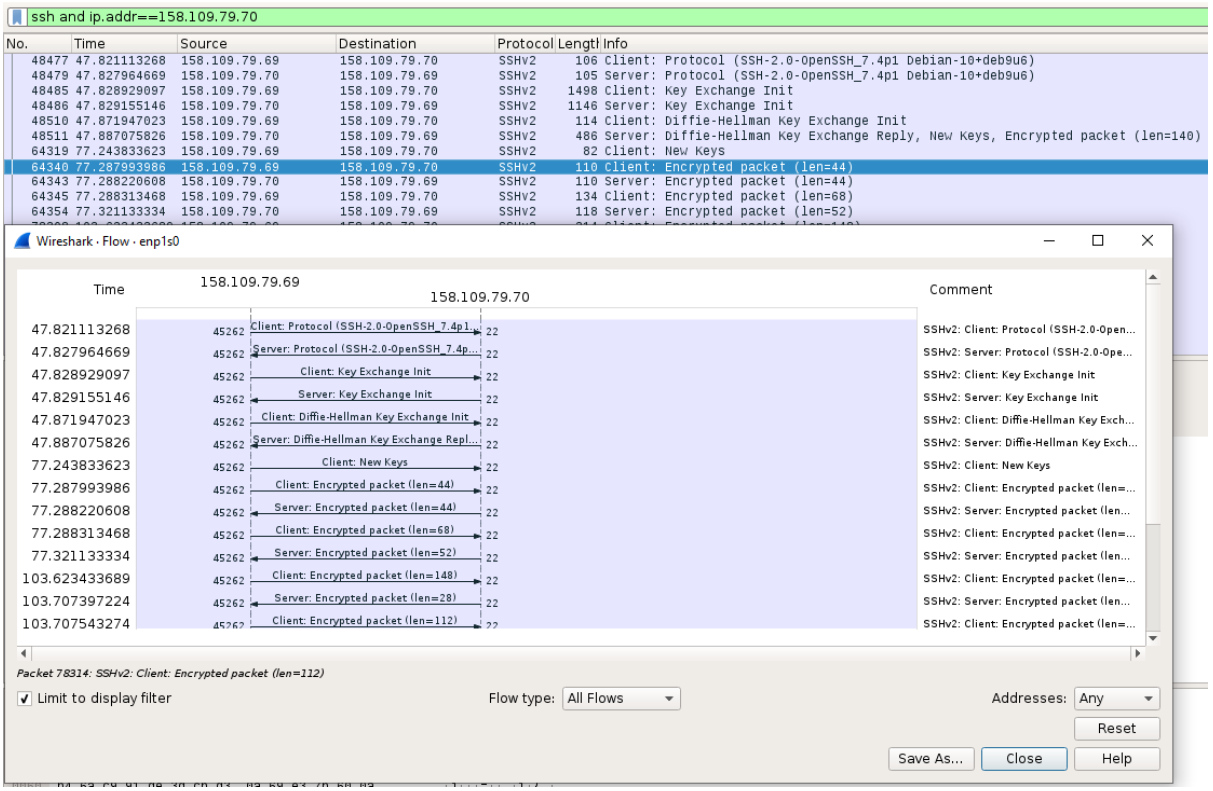
Josep Douton Martin - 1397823
Julio Velásquez Cárdenas - 1397896
Grupo: itx-e3

3.3. Tráfico SSH

1. (1 punto) Filtre en Wireshark para que capture solo tráfico SSH. Conéctese a través de SSH a otra maquina con ssh usuario@DEIC-DCX. Una vez iniciada la sesión detenga la captura. Utilice la herramienta Flow Graph, haga una captura del gráfico desde el inicio de la negociación TCP hasta el envío de los 3 primeros datos cifrados. Resalte el método (dentro de un canal público) de intercambio de llaves utilizado.

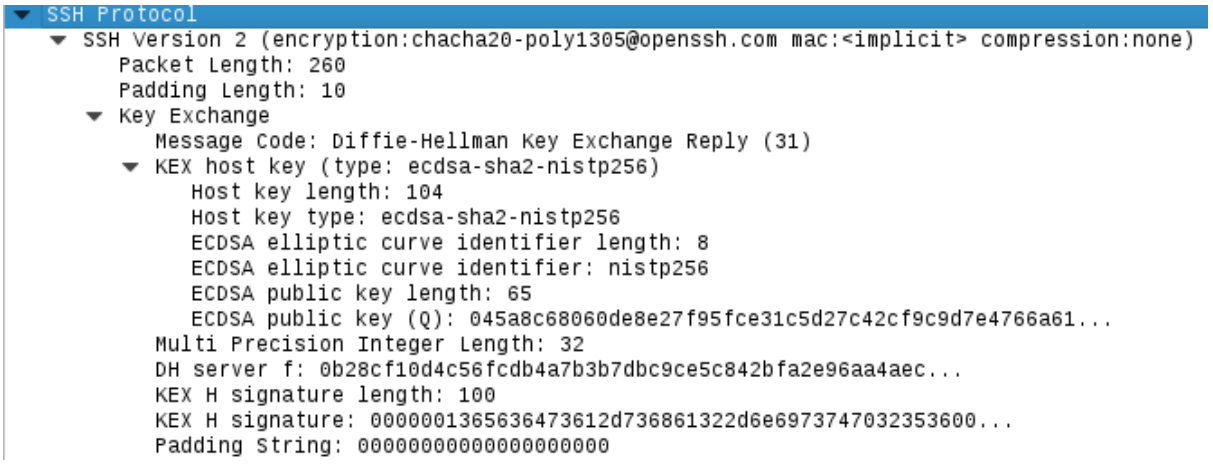


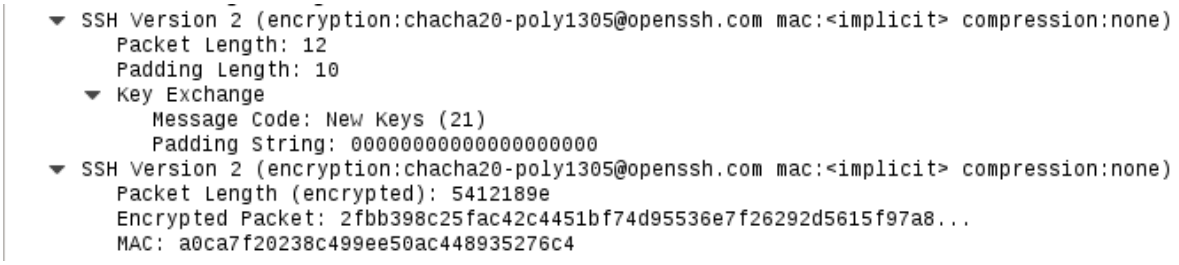
2. (1 punto) ¿Cuál es el puerto local que se utiliza para la conexión SSH?

Cómo se puede observar el ssh utiliza el puerto 22 y el de vuelta 45262.

Josep Douton Martin - 1397823
Julio Velásquez Cárdenas - 1397896
Grupo: itx-e3

3. (1 punto) ¿Qué versión de SSH, cifrado, Message Authentication Code (MAC) y compresión se negoció?. Colocar una imagen





3.4 Tráfico HTTPS

A continuación vamos a crear un servidor que sirve una página Web en HTTPS y utilizaremos Wireshark para ver como se puede capturar esta información y qué información podemos obtener.

1. Comenzaremos creando una llave RSA y un certificado para el servidor, abrimos una consola y los generamos con el siguiente comando:

openssl req -new -x509 -out server.crt -nodes -keyout server.pem -subj /CN=localhost

Con este pedido solicitamos a OpenSSL que nos genere un certificado (-NEW) autofirmado (-x509) que se guardara en el archivo (´ -out server.crt), pedimos que nos genere una llave privada (-keyout server.pem) y que no la cifra (-nodes) y especificamos el CommonName del certificado (-subj / CN = localhost).

2. Una vez que tengamos el certificado y la llave privada, podemos abrir el servidor con el siguiente comando:

openssl s server -tls1 2 -www -cipher AES256-SHA -key server.pem -cert server.crt

Josep Douton Martin - 1397823
Julio Velásquez Cárdenas - 1397896
Grupo: itx-e3

En este caso estamos requiriendo a OpenSSL que nos abra un servidor que utiliza AES para cifrar (-cipher AES256-SHA) que utilizara la llave privada (-key server.pem) y el certificado (-cierto server.crt) que acabamos de crear y que nos responderá a las peticiones con un mensaje de estado (-www).

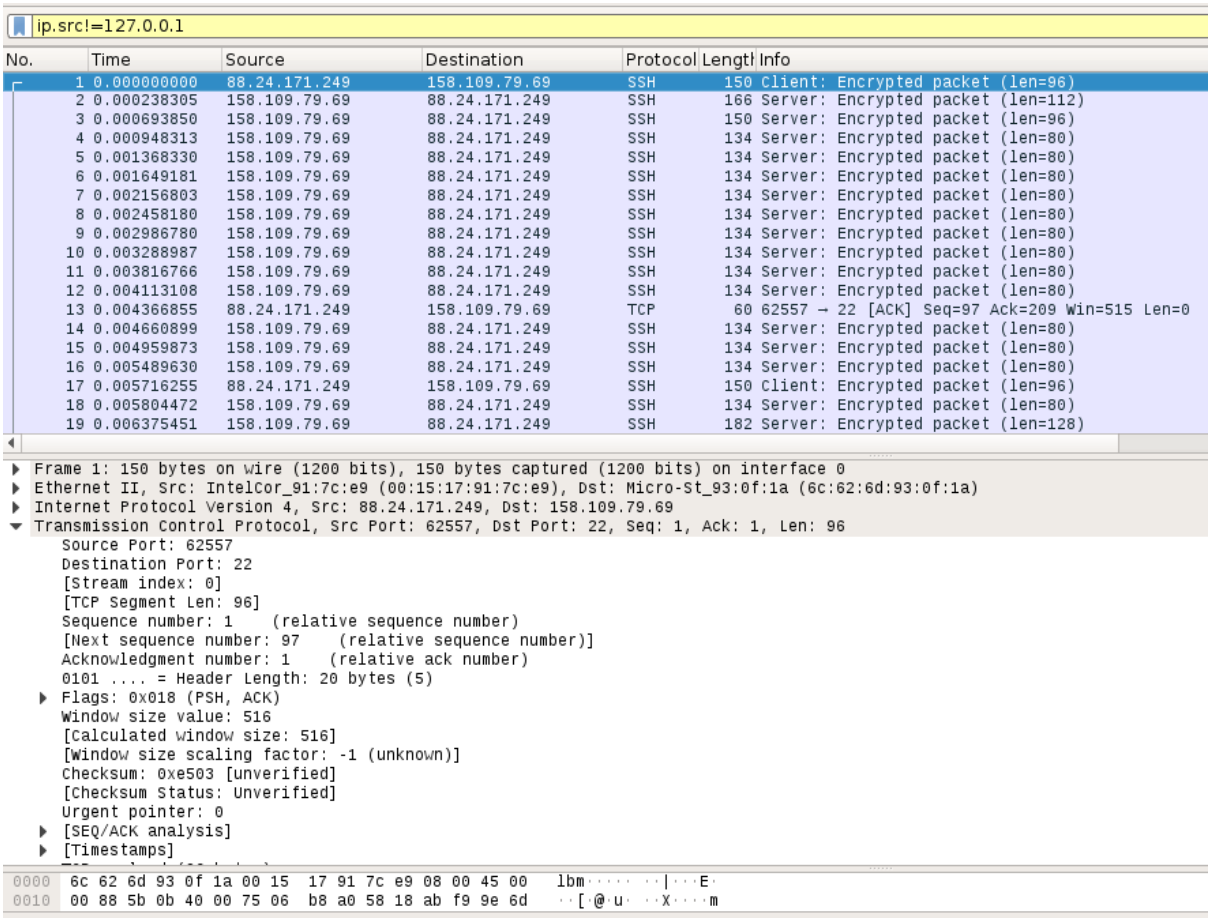
3. Ahora ejecutamos Wireshark para capturar el tráfico que se genere hacia y desde localhost. Se recomienda aplicar como filtro de pantalla ”ssl”

4. Una vez iniciemos la captura del tráfico, en una nueva consola haremos la petición Web con el comando:

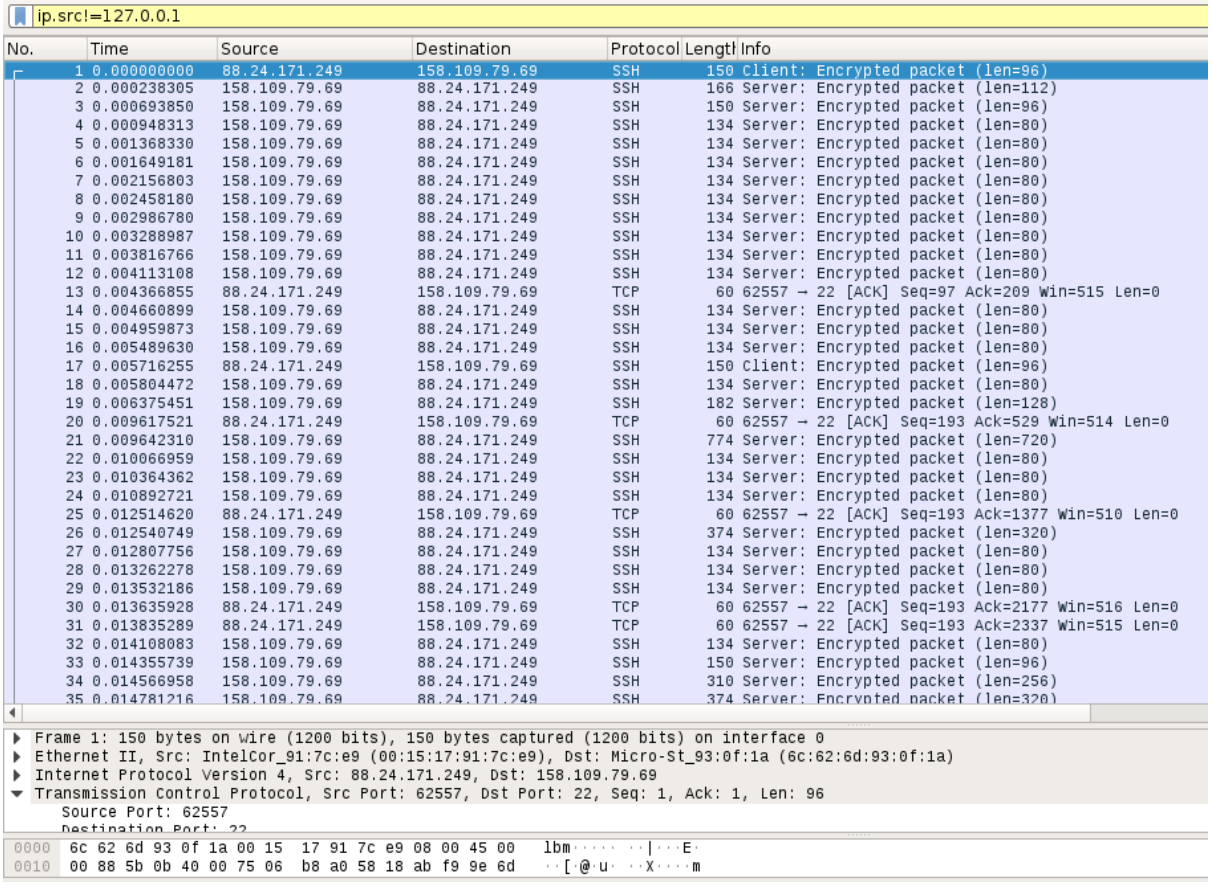
printf ’GET / HTTP/1.0\r\n\r\n’ | openssl s_client -ign_eof

5. (1 punto) Explica lo que has obtenido en esta captura. Colocar una imagen

Al no tener la clave para el descifrado SSL, Wireshark ve todos los paquetes cifrados sin saber la información que contienen.



Josep Douton Martin - 1397823
Julio Velásquez Cárdenas - 1397896
Grupo: itx-e3



6. Ahora le diremos a Wireshark cual es la llave privada que se está utilizando en esta comunicación, para hacer esto iremos a " Edit"→"Preferences", dentro de Protocolos buscaremos SSL y modificaremos la RSA keys lista, con los parámetros: IP address - 127.0.0.1, Puerto - 4433, Protocolo - http y Key File - server.pem. (En caso de que no se descifran los mensajes cambiar la IP address por ::1). Finalmente al darle doble click sobre uno de los paquetes cifrados se habilitará una nueva pestaña Decrypted SSL al lado de Frame

7. (1 punto) ¿Qué cambios se han producido en los paquetes? ¿Cuál es la respuesta del servidor a la petición GET?. Colocar la imagen obtenida al ingresar a Follow SSL Stream.

Como ahora Wireshark tiene acceso a la clave de cifrado puede desencriptar los paquetes que ha capturado y leer su contenido. Gracias a eso es capaz de ver los paquetes con los mensajes de "Client Hello" y "Server Hello" por ejemplo.