# Práctica/Laboratorio de TCP

Para la captura dada analizar con el siguiente cuestionario utilizando una herramienta como wireshark.

1. ¿Cuántos intentos de conexiones TCP hay en la captura?  
   En la captura obtuvimos un único intento de conexión, mediante la utilización del filtro “tcp.flags.syn== 1 && tcp.flags.ack==0”. Usamos estos flags, ya que la combinación de SYN en 1 y ACK en 0. Nos devuelve los intentos de conexión por parte del cliente. Si hubiéramos desestimado el flag de ACK el resultado sería además del intento de conexión de lado cliente, la respuesta del servidor al cliente con los flags SYN=1 y ACK=1 (confirmando la solicitud de conexión)
2. Cuales son la fuente y el destino (IP:port) para c/u?

Para la conexión obtenida en el punto anterior:

Fuente= 10.0.1.10:49357

Destino= 10.0.4.10:5001

1. ¿Cuántas conexiones TCP exitosas hay en la captura? Cómo se identifican las exitosas de las no exitosas, que flags se encuentran en estas?

Hay una sola conexión exitosa en la captura (Hay unicamente una conexión en toda la fucking captura). Identificamos la conexión exitosa, ya que se realizo correctamente el saludo de 3 vias.

Dada la primera exitosa responder:

1. ¿Quién inicia la conexión, quien sería el servidor y quién el cliente? Qué flags se ven activados? En que segmentos se ve el 3-way hand-shake?

La conexión es iniciada por 10.0.1.10:49357.

El servidor es 10.0.4.10:5001.

El cliente es 10.0.1.10:49357.

Flags activados en el intento de conexión **SYN**. Los segmentos correspondientes son **3, 4, 5**

1. Qué ISNs se intercambian?

ISN “cliente”: 4293989133

ISN “servidor”: 2543122073

1. Qué opciones se negocian. Qué significa c/u? ¿Cuál es el MTU negociado?  
   Las opciones negociadas son:
   * MSS:
   * SACK Permitted:
   * TimeStamps
   * NOP
   * Windows Scale

El MTU del “cliente” es 1460 bytes.

El MTU del “servidor” es 660 bytes.

1. Quién es el que envía la mayor cantidad de datos (IP:port)?

Agregamos una nueva columna (Cumaltive Bytes) en el display principal de wireshark. Que nos entrega la sumatoria del tamaño de los datos enviados por parte del cliente como asi tambien la del servidor. Una vez filtrados los datos por ip origen, podemos concluir que el que envia mayor cantidad de datos en la captura es (10.0.1.10:49357).

10.0.1.10:5001 -> 904488 bytes enviados (cabeceras incluidas)

10.0.4.10:49357 -> 67224 bytes enviados (cabeceras incluidas)

Otra forma de verlo, contando solamente bytes de carga util (sin cabeceras ni nada de esa mierda) Si el cliente inicia con seq= 4293989133 y finaliza con seq= 4294775591 y se que incrementa el número de secuencia en 1 por cada byte de datos transmitidos. Facilmente puedo concluir que la diferencia de estos num de secuencia seria el numero de bytes enviados. En este caso el cliente envio 786458 B (aca estaria incluido el syn) = 0.750025 MB .

Por otro lado el servidor nunca hace envio de datos de carga util al cliente, solamente secuencias de confirmacion.

1. Identificar primer segmento de datos (origen, destino, tiempo, número de fila y número de secuencia TCP).

Primer segmento que envia datos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num fila | Tiempo | Origen | Destino | Num seq |
| 6 | 0.064722 | 10.0.1.10:49357 | 10.0.4.10:5001 | 4293989134 |

* 1. ¿Cuántos datos lleva?

Longitud de datos: **24**

* 1. ¿Cuándo es confirmado (tiempo, número de fila y número de secuencia TCP)?

|  |  |  |
| --- | --- | --- |
| Num fila | Tiempo | Num seq |
| 14 | 0.095270 | 2543122074 |

* 1. ¿La confirmación, qué cantidad de bytes confirma?

Ack =4293989158 **PREGUNTAR**

1. Control de Flujo:
   1. ¿Se activa en algún momento el mecanismo de control de lujo?

NO. En ningún momento de la captura se activa el mecanismo de control de flujo. Esto ocurriría si algún buffer de recepción tanto del cliente como del servidor se viera desbordado. (En esta caso solo tiene sentido observar el tamaño del buffer receptor del “servidor” ya que es el único que esta continuamente recibiendo datos). En caso de desbordar algún buffer receptor (claro indicio de activación de control de flujo) wireshark indicara este evento con el mensaje de “ZeroWindow”.

El control de flujo es un servicio de control de velocidades (adapta la velocidad a la que el emisor está transmitiendo frente a la velocidad a la que la aplicación receptora está leyendo).

* 1. Indicar donde (tiempo, número de fila y número de secuencia TCP) ¿y a que se debe?

No tiene sentido responder.

* 1. ¿Cuánto tiempo parece durar?

Idem b.

* 1. ¿Cuál es el número de ventana que desactiva el mismo?

Idem b.

* 1. ¿Qué otros datos se pueden obtener?

Idem b.

1. Control de Congestión:
   1. ¿Se encuentra en la red indicios de congestión?

Si. Uno de los indicios de congestión es la perdida de segmentos, si un segmento se pierde tcp debe retransmitir el segmento.

* 1. ¿Cómo se detectan?, Indicar un número de segmento perdido.

Los segmentos perdidos deben ser retrasmitidos.

Un caso de pérdida de paquetes es la perdida de segmento por RTO (retransmisión típica), y otra puede ser la retransmisión por la llegada del tercer ack duplicado perteneciente a ese segmento, por lo que se puede asumir con grandes probabilidades que el paquete se perdió y inicia una fast-restransmission antes que llegue a time out, esto se hace para mejorar el rendimiento de la red.

Un numero de segmento perdido en la captura puede ser por ejemplo el Seq= 4294026094 perteneciente al frame numero 141.

* 1. En que momento se ve la primera retransmisión (tiempo y número en el analizador)?

Primera retransmisión (fast-restransmission)

|  |  |
| --- | --- |
| Tiempo | Num fila |
| 0.262890 | 92 |

* 1. ¿Cuántos segmentos se re-transmiten?

En la captura se realizan 38 fast-retransmission.

1. ¿Quién inicia el cierre de la conexión? Qué flags se utilizan? En que segmentos se ve esta (tiempo, número de fila y número de secuencia TCP) ?

El cierre de conexión es iniciado por el cliente (10.0.1.10:49357), este ese segmento los flags utilizados son:

* FIN: Indica la petición de fin de conexión.
* PSH:
* ACK:

|  |  |  |
| --- | --- | --- |
| Tiempo | Numero de fila | Numero secuencia |
| 10.360495 | 2224 | 4294775182 |

1. ¿El RTT entre que valores oscila?

El rtt (Round Trip Time) de la captura oscila entre los valores min= 30ms y max= 405ms

1. ¿El BW digital alcanzado, cual parece ser?

El BandWidth (ancho de banda) alcanzado 688 kbits/seg

1. ¿Qué otros datos puede obtener de la captura sobre el flujo analizado?