HDC

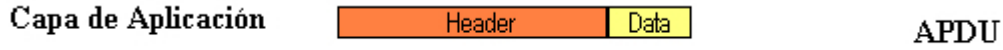
Bienvenidos a otra edición de “Hacking desde cero by Roadd Dogg”:D Hoy analizaremos y comprenderemos la teoría detrás de los paquetes de datos.

Empecemos teniendo algo en claro. **Para pasar información de un nodo a otro**, no podes pasar una cosa sin formato, como se te cante y al aire, esperando que el receptor sepa para quien es el mensaje y poder decodificiarlo. Además, **generalmente se utiliza cierta separación lógica, previamente acordada, para que sea más comprensible el mensaje.** Así como en el lenguaje natural utilizamos palabras, en español, e intentamos de hacer que el receptor sepa que el mensaje es para él.

Y esto ¿Cómo funciona en las **computadoras**?

Vamos a ver un ejemplo con un mensaje cualquiera. Éste tendrá **información** y se llamará **Unidad de datos.**

hora digamos que lo escribiste en una **aplicación** como Whatsapp. Entonces, la aplicación le mete un **encabezado** o **header** que le dice que es de este mismo programa y se une a los bytes del mensaje. Y de ahí viaja el llamado **APDU**, para la capa de presentación.



Esta capa, agarra el APDU y le encastra **otro header** más, **delante del anterior** y generalmente formatea los datos. Los **codifica** o los **comprime** para la comunicación. De aquí sale el **PPDU**.

La **capa de sesión** también coloca su **header** (todos quieren meter su lindo header) delante del anterior nombrado y es la que se encarga de que el mensaje **llegue** a destino **y reenviarlo** si es necesario. Llamemos a éste, **SPDU**.

Llega felizmente a la **capa de transporte** donde se transformará en un **TPDU**. Para esto, adivinen... sí, le pondrá su querido **header**. Se acuerdan que yo les dije que en esta capa se manejaba el control de puertos, ¿No? Bueno,lo hace. Entonces ya suponen **qué vendrá en el header**. ¡Claro! El **número de puerto** de donde sale y hacia que puerto debería llegar.

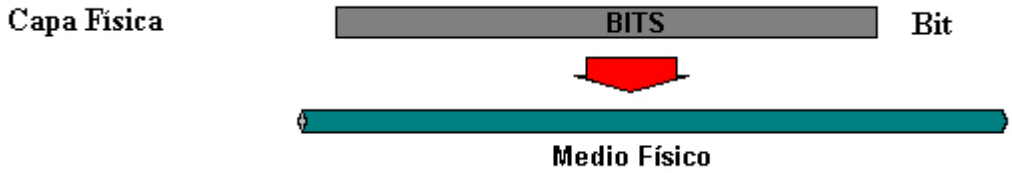
Aquí se va a formar el verdadero y famoso **paquete de datos.** En la **capa de red**, se le coloca un **header** que irá delante de todo y un **tail** que tomará parte detrás de todo el TPDU.

En esta capa, que maneja las **direcciones IP,** se verán las IP destino y llegada. Claro que esta información se le coloca al paquete.

Sigamos más aún porque eso no es lo unico que interesa.

El paquete de datos se pasa a la **capa de enlace de datos donde** también se coloca un **header** y un **tail**, para luego llamarlo **trama**. Aquí se coloca la información referida a la **MAC address.**

Y luego la trama, llegará a la **capa física** para que ésta **transforme** todos esos números binarios en impulsos eléctrico u ondas electromagnéticas o **la forma que sea que tiene que tomar para poder mandar, físicamente, la información a través de el medio requerido.**

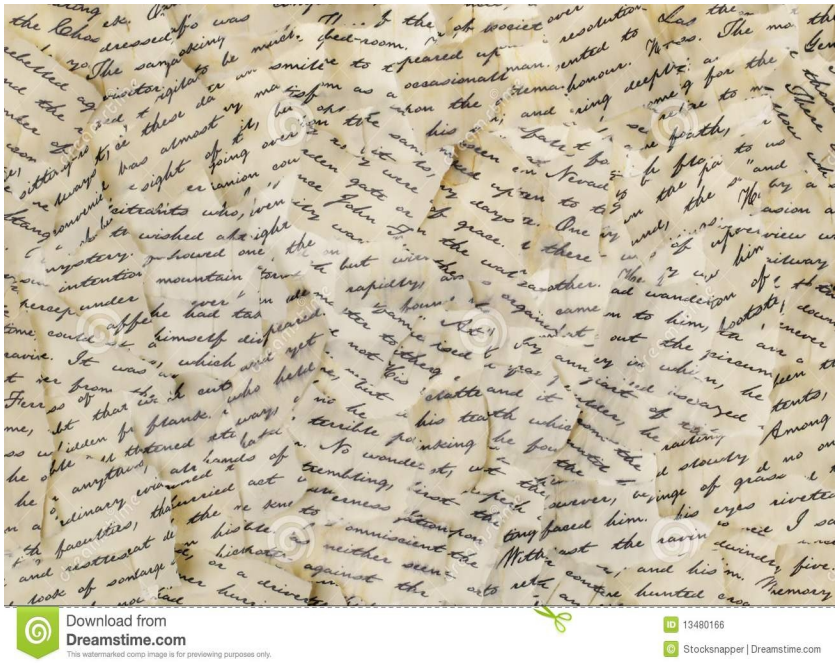


“**Aaaaaaaaaah. Ahora entiendo bastante más, el uso del modelo OSI.”**

Claro, ahí vamos viendo como todo se relaciona y qué funcionalidad tiene cada capa.

Y los mensajes, pueden llegar a ser larguísimos y por eso **no se envían en un sólo paquete**, sino que se envían muchos, de **manera fragmentada**, para que **luego** se **reconstruyan** en el próximo nodo. Muchas veces cuentan con un **checksum** -como un hash- para **conocer** si el paquete llegó en buen **estado** o llegó averiado -y si es una comunicación TCP, pedir el paquete nuevamente-.

También se manda un orden con el que los fragmentos se mandan, así no importa el tiempo en el que llegue cada uno, después el sistema puede reconstruir todo el mensaje sin problemas ni cosas raras.



“**Otra cosa. Los paquetes SYN, FIN, ACK, etc.”**

Ahí entiendo. Los paquetes SYN y demases (que se llaman **paquetes TCP**), mandan un mensaje con:

-Un header TCP que dice cuál es el largo del mensaje.

-3 bits vacíos.

-9 bits que corresponden a flags. Los flags son bits (un bit cada flag) que son usados para estado y control. Y esos flags son:

-**NS**: De protección.

-**CWR**: Cuando recibe un paquete con el flag ECE en 1, responde con un paquete y este bit activado para el control de congestión.

-**ECE**: Da indicaciones de congestión (en 1, congestionado. En 0, no).

-**URG**: Flag de paquete urgente.

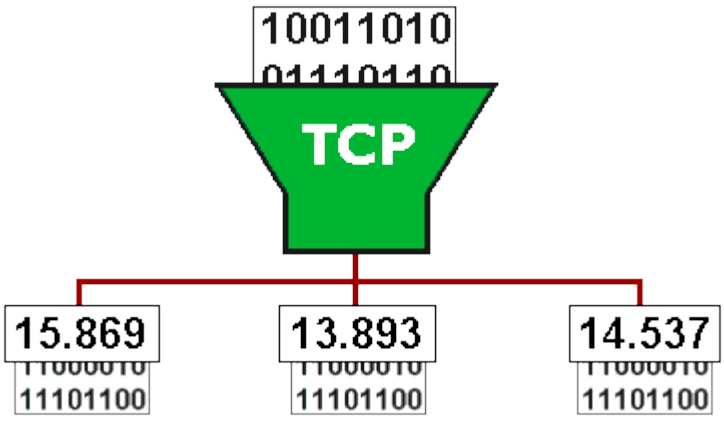
-**ACK**: Ya conocemos cuándo se usa este flag ;).

-**PSH**: Los datos tienen que llegar urgente a la aplicación si este flag está en 1.

-**RST**: Resetea o no acepta la conexión.

-**SYN**: Famoso para sincronizar;).

-**FIN**: Finaliza la conexión.



Entonces, un paquete SYN, tendrá un header con el largo, 3 bits vacíos, y todos los flags en cero, menos el SYN que estará en 1 y así el dispositivo que recibe el mensaje lo entenderá.

Bueno dejemos acá así no les machaco demasiado el cerebro. Creo que no tienen nada que temerle a las comunicaciones. ¿Verdad? Siéntanse seguros.

Cualquier cosa pueden mandarme mail a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente:

1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

Roadd.

Este tutorial puede ser copiado y/o compartido en cualquier lado siempre poniendo que es de mi autoría y de mis propios conocimientos.