**PROTOCOLOS Y PUERTOS**

Un protocolo es como tener una determinada manera de realizar una comunicación, para que ambas partes se entiendan. El protocolo más importante es el protocolo IP, de este derivan otros. Hay protocolos para cada capa del modelo OSI o del modelo TCP/IP

En la capa de transporte tenemos donde se organiza que puerto va a qué servicio (Esto lo hace una organización llamada IANA). Aquí te dice que servicio DEBERÍA de correr en que puerto, pero no es una obligación, esto se puede alterar.

Los puertos con sus servicios más relevantes son (Es importante saber esto):

* 21 -> Se relaciona con el protocolo FTP. Dando servicios como ***vsftp***, ***ProFTPD***, ***FileZilla Server.***
* 22 -> Se relaciona con el protocolo SSH. Dando servicios como ***OpenSSH*** o ***PuTTY***.
* 23 -> Se relaciona con el protocolo Telnet (versión antigua de SSH), es usado para conectarnos a otra máquina de forma remota. Es bastante inseguro, pero hay empresas que lo siguen usando.
* 25 -> Se relaciona con el protocolo SMTP. Dando servicios como ***Microsoft Exchange***, ***Postfix***, ***Sendmail***. SMTP es un protocolo utilizado en el envío y recepción de correos electrónicos.
* 53- > Se relaciona con el protocolo DNS. Dando servicios como ***BIND***, ***Microsoft DNS***, ***Unbound***.
* 80 y 443 -> Se relaciona con el protocolo HTTP y HTTPS (respectivamente). Dando servicios como ***Apache HTTP Server***, ***Nginx***, ***Microsoft IIS***.
* 110 y 143 -> Se relaciona con el protocolo POP3 e IMAP. Dando servicios como ***Dovecot*** y ***Exchange***.
* 135 -> Se relaciona con el protocolo RPC. Dando servicios como ***DCOM.***
* 139 y 445 -> Se relaciona con el protocolo SMB y SAMBA respectivamente.
* 161 -> Se relaciona con el protocolo SNMP. Dando servicios con ***Net-SNMP*** (Linux) o ***SNMP*** (Windows).
* 1433 -> Se relaciona con el protocolo Microsoft SQL Server.
* 3306 -> MySQL.
* 5432 -> PostgreSQL.
* 3389 -> RDP.
* 6379 -> Redis.
* 27017 -> MongoDB.
* 9200 y 9300 -> ElasticSearch.

**ARP (Address Resolution Protocol).** Protocolo de resolución de direcciones. Relaciona en una red una dirección IP de un equipo con su MAC. Esta información la almacena en una tabla, la cual se puede consultar. Este protocolo no tiene autenticación, con lo cual un intruso podría atrapar el mensaje mediante Spoofing.

**DHCP.** Protocolo encargado de asignar una IP a un nuevo equipo en la red. Este protocolo utiliza el proceso DORA, el cual consta de 4 pasos:

* Discover: El cliente envía un mensaje sondeando que servidores DHCP están disponibles.
* OFFER: Un servidor que recibe el mensaje, responde a este con un mensaje DHCPOFFER. Contiene una oferta con una dirección IP y demás datos como la máscara, puerta de enlace y servidores DNS.
* REQUEST: El cliente le responde al servidor solicitando la oferta formal del servidor. Este paso se hace de forma BROADCAST, para que en caso de que haya varias ofertas de varios servidores, les indiquemos a todos que oferta hemos aceptado.
* AKNOWLEDGEMENT: El servidor confirma la solicitud con un mensaje DHCPACK y le asigna la IP.

**DNS.** Protocolo que relaciona el nombre de una web con su IP.

**ICMP (Internet Control Message Protocol).** Protocolo diseñado exclusivamente para transmitir mensajes de control y de error entre nodos en una red. Se usa para mantenimiento de una red(capa 3).

Una conexión por ICMP consta generalmente de 3 pasos.

* Generamos un mensaje ICMP: Ocurre cuando se encuentra un problema al procesar paquetes IP.
* Encapsulamiento del mensaje ICMP: Se encapsulan en un paquete estándar de IP. De esta forma los equipos lo identifican como un paquete ICMP.
* Transmisión del mensaje: Este se transmite a través de la red hacia el dispositivo en cuestión. (por ejemplo, si hay un receptor inalcanzable, el mensaje ICMP se devolverá al emisor).

Estos mensajes se generan generalmente con el comando ping.

En la respuesta el TTL nos puede indicar que SO tiene el receptor.

TTL = 64 APROX (LINUX)

TTL = 128 APROX (WINDOWS)

TTL = 255 APROX (Dispositivos Routers y Switches)

Estos valores se pueden modificar.

Al utilizar la capa 3, no utiliza puertos.

**MODELO OSI**

**CAPA 1 (Capa física):**

Es la más externa, vendría a ser el hardware que se utiliza para la transmisión de datos, por ejemplo, el cable de fibra óptica, routers, hubs...

Lo más destacable de esta capa es la transformación de las señales digitales en señales físicas, como tensiones, por ejemplo.

**CAPA 2(Enlace de datos):**

Esta se usa fundamentalmente para detectar fallos y corregir errores que se pueden haber producido en la capa física.

En esta capa los datos son llamados tramas y podemos encontrar la MAC.

Capa que controla el flujo para que el receptor no se sobrecargue con demasiados datos.

**CAPA 3 (Capa de red):**

Enrutador, realiza el recorrido que van a tener los datos para llegar a su destino. Los enrutadores intercambian información para determinar la mejor ruta.

En esta capa los datos pasan a llamarse paquetes y son encapsulados (se les añade una cabecera o se complementa la que hay con información de IP de envío e IP del receptor) para su envío.

El protocolo IP es el fundamental en esta capa.

**CAPA 4(Capa de transporte):**

Se establecen las conexiones para realizar el envío. En esta etapa los protocolos fundamentales son TCP y UDP, según el protocolo los datos se llaman segmentos o datagramas respectivamente.

El protocolo TCP es más lento, pero se asegura que los datos lleguen en orden y correctos. En UDP sacrificamos calidad por velocidad.

Para el envío de datos, en esta etapa se generan encabezados para cada segmento con información necesaria para su entrega, como número de secuencia, puertos de origen y destino, etc…

**CAPA 5(Capa de sesión):**

Cuando se ejecuta una comunicación es como si se abriese una sesión y sirve de “juez” para garantizar que la comunicación se realiza de forma correcta, controlando quien “habla” y quien “escucha”

En esta capa se gestiona y se administra el inicio de la sesión hasta que termina.

**CAPA 6(Capa de presentación):**

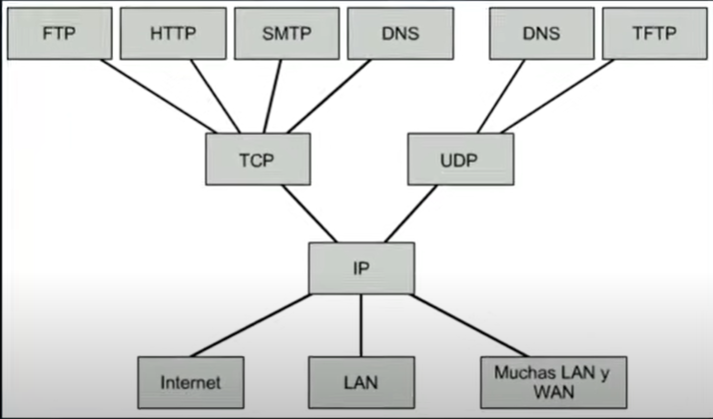
Hasta esta capa los datos no son legibles para nosotros y en esta capa se transforman. También sirve de interprete entre los dispositivos, como si de un lenguaje de programación se tratase (ASCII).

En esta capa se genera la clave para el cifrado de los datos, el cifrado y el descifrado de los datos.

**CAPA 7(Capa de aplicación):**

En esta reside toda la interacción entre usuario y el resultado final de los datos recibidos.

En esta etapa nos encontramos protocolos como HTTP/HTTPS, SMTP, FTP, o DNS.



**NOTA:** Las capas en verdad funcionan como 1 sola. Pero las separamos en 7 capas para su mayor comprensión.

**NOTA2:** Hay una 8º capa que son los humanos, esta capa es la más vulnerable.

**TCP**

Profundizando un poco más en este protocolo, vamos a explicar el mecanismo de establecimiento de conexión (apretón de manos de TPC).

* Un equipo inicia una conexión enviando un segmento TCP (SYN). Dentro de este segmento se incluyen:
  + Una flag SYN establecida en 1, para indicar al otro equipo que es una solicitud de sincronización para establecer una conexión.
  + Número de secuencia (ISN), es un número aleatorio.
* El equipo receptor (generalmente un servidor) responderá con otro segmento TCP (SYN-ACK). Dentro se incluirá:
  + Una flag SYN establecida en 1 si está de acuerdo en establecer la conexión.
  + Una flag ACK establecida en 1 para reconocer la recepción del segmento anterior.
  + Número de secuencia (seq Number), es un número aleatorio.
  + Número de acuse de recibo (Ack number), es el ISN + 1.
* El equipo emisor realiza un nuevo segmento (ACK) donde se incluye:
  + Flag ACK establecida en 1 para confirmar que hemos recibido el segmento SYN-ACK.
  + El Ack number que nos envió.
  + El Seq Number + 1.