**PRACTICA 3.8**

**NIVEL DE TRANSPORTE: TCP – UDP**

**1. Abre con el programa wireshark la captura udp.cap en la que se muestra una**

**comunicacion UDP. Contesta a las siguientes preguntas:**

**a) .Cuales son las direcciones IP y los puertos asociados a la comunicacion?**

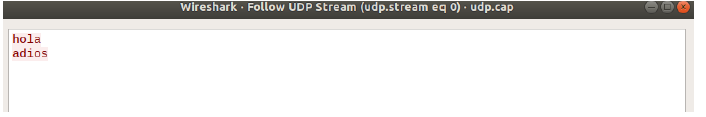
Src: 10.0.0.2 Dst: 11.0.0.2

Src Port: 32768 Dst Port: 33000

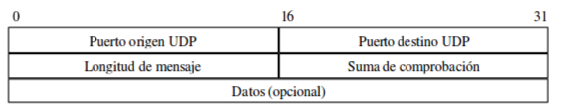
**b) .En que categoria se encuentran los puertos utilizados?**

Se encuentra en la categoría Registrados porque se encuentra entre 1024-49151

**c) Comprueba los datos enviados desde Analyze → Follow UDP Stream.**



**d) Indica que campos tiene una cabecera UDP**



**2. Abre con el programa wireshark la captura tcp.cap en la que se muestra una**

**comunicacion TCP. Contesta a las siguientes preguntas:**

**a) .Cual es la direccion IP y el puerto del cliente TCP y la direccion IP y el puerto del**

**servidor TCP?**

Cliente:

IP: 10.0.0.2 Puerto: 60709

Servidor:

IP: 11.0.0.2 Puerto: 34000

**b) .En que categoria se encuentran los puertos utilizados?**

El cliente se encuentra en el rango Dinámico/Privado (1024-49151)

El servidor se encuentra en el rango Registrados (49152-65535)

**c) Comprueba los datos enviados desde Analyze → Follow TCP Stream.**



**3. Segun lo observado en las dos preguntas anteriores explica resumidamente que**

**diferencias existen entre TCP y UDP.**

**TCP:**

-Establece conexión entre ordenadores antes de transmitir datos.

-Está orientado a conexión.

-Es lento.

-Tiene una confiabilidad altamente fiable.

-Requiere reconocimiento de datos y si el usuario quiere, se puede volver a

transmitir.

**UDP:**

-Envía los datos directamente al destino sin confirmar si está listo para recibir.

-No orientado a conexión.

-Confiabilidad no fidedigna.

-Es rápido.

-No requiere conocimientos ni retransmitir datos perdidos.

**4. Utilizando wireshark accede a una pagina web, analiza la primera conexion**

**HTTP. Rellena los siguientes datos:**

**• Puerto de origen TCP:**

50730

**• Clasificacion del puerto de origen:**

Registrado

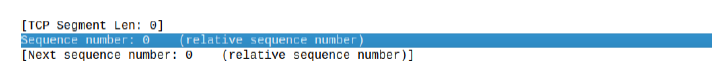
**• Puerto de destino TCP:**

443

**• Clasificacion del puerto de destino:**

Bien conocidos

**• .Cual es el numero de secuencia relativa establecido?**



**5. Selecciona la respuesta a la peticion HTTP analizada en el ejercicio anterior, en**

**wireshark selecciona “Go → Next Packet in Conversation”**

**• .Cuales son los valores de los puertos de origen y destino?**



**• .Cuales son los numeros de acuse de recibo y de secuencia relativa**

**establecidos?**



**6. Utilizando wireshark accede a una pagina web, analiza las consultas y**

**respuestas DNS. Rellena los siguientes datos:**

**• Direccion MAC de origen y de destino:**



**• Direccion IP de origen y de destino:**



**• Puerto de origen y de destino:**



**7. Vamos a crear una aplicacion cliente/servidor que utiliza sockets. Crea el**

**siguiente fichero llamado “servidor.py”**

*#!/usr/bin/python3*

*import socket*

*IP\_SERVIDOR = "192.168.25.122” /\* IP del usuario Axel, donde se aloja el servidor\*/*

*PUERTO\_SERVIDOR = 4444*

*#Crea el socket*

*s = socket.socket()*

*#Espera conexiones en la IP y puerto establecidos*

*s.bind((IP\_SERVIDOR, PUERTO\_SERVIDOR))*

*print ("Esperando conexiones en el puerto "+ str(PUERTO\_SERVIDOR)+"...")*

*# Espera una conexión y envía confirmación*

*s.listen(1)*

*socket\_cliente, addr = s.accept()*

*print("Conexión recibida de: " + str(addr))*

*envia = "Conexion con el servidor establecida."*

*socket\_cliente.send(bytes(envia,"utf-8"))*

*while True:*

*recibido = socket\_cliente.recv(1024)*

*if (recibido.decode() == "quit"):*

*print ("Conexión cerrada.")*

*break*

*print("Recibido: "+recibido.decode())*

*socket\_cliente.close()*

*s.close()*

**Ahora crea el siguiente fichero llamado “cliente.py”:**

*#!/usr/bin/python3*

*import socket*

*IP\_SERVIDOR = "192.168.25.122”*

*PUERTO\_SERVIDOR = 4444*

*#Crea el socket*

*s = socket.socket()*

*print ("Intentando conectar "+str(PUERTO\_SERVIDOR)) con "+IP\_SERVIDOR+" en el*

*puerto "+str(PUERTO\_SERVIDOR))*

*# Conexión a la IP y puerto del servidor*

*servidor = (IP\_SERVIDOR, PUERTO\_SERVIDOR)*

*s.connect(servidor)*

*# Recibe hasta 1024 bytes*

*recibido = s.recv(1024)*

*print(recibido.decode())*

*print ("Enviar mensajes al servidor (escribe 'quit' para terminar): ")*

*mensaje = ""*

*while (mensaje != "quit"):*

*mensaje = input("> ")*

*s.send(bytes(mensaje, 'utf-8'))*

*s.close()*

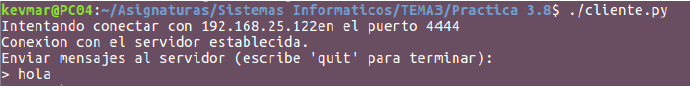
**Establece permiso de ejecucion a los ficheros creados “chmod +x \*.py” y ejecuta**

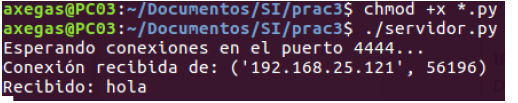
**cada uno en un terminal distinto. Comprueba que funciona. Modifica los ficheros**

**para establecer conexion con el equipo de algun companero. .Que es un socket?**

**.Que informacion necesitan los sockets para establecer una conexion? .Que**

**puertos se usan en el cliente para establecer la conexion?**





Socket: Es un segmento TCP. Queda definido por un protocolo de transporte, un nº de puerto y una dirección IP. Para establecer una conexión entre dos equipos, tiene que conocer el puerto de origen, (56196 en nuestro caso) el puerto destino (4444 en nuestro caso), la dirección IP destino y la dirección IP origen.

**8. Con la aplicacion cliente/servidor del ejercicio anterior en ejecucion obten una**

**captura de pantalla de todas las conexiones establecidas por tu equipo con el**

**comando “netstat”. Busca las conexiones relativas a tu aplicacion .Que protocolos**

**de la capa de transporte se estan utilizando? .Que numeros de puertos se estan**

**usando en las conexiones? .Como se clasifican esos puertos?.**

* Se está usando el protocolo TCP
* Se utilizan los puertos 4444(servidor-puerto registrado) y 56212(cliente-puerto

dinamico/privado)

