**Laboratorio 1 – Protocolos de la capa de**

**Transporte y Aplicación**



**Jhossef Nicolas Constain Nieves**

**Profesor:**

**Edwin Ferney Castillo Quintero**

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Departamento de Sistemas** **Redes**

**Popayán, Febrero 2024**

**Introducción:**En el análisis de la comunicación de red entre un cliente y varios servidores utilizando los protocolos HTTP y FTP, es crucial comprender los detalles de la interacción entre las máquinas y los protocolos involucrados. Este análisis se realiza utilizando herramientas como Wireshark, que permite capturar y examinar el tráfico de red en detalle. En este caso, se ha capturado el tráfico relacionado con el acceso a servicios web específicos, como Univirtual, la Recarga Biblioteca, y un servidor HTTP.

Los datos capturados incluyen información como las direcciones IP del cliente y del servidor, los números de puerto, los protocolos de transporte utilizados (TCP en este caso), y detalles de las solicitudes y respuestas HTTP. A través de este análisis, podemos entender cómo se establecen y finalizan las conexiones, qué datos se transfieren entre el cliente y el servidor, y cómo se manejan diversas interacciones entre ellos.

En esta actividad, se abordarán preguntas clave relacionadas con la identificación de las computadoras involucradas, el proceso de establecimiento de conexiones TCP utilizando el método de tres vías (three-way handshake), la finalización de sesiones HTTP y FTP, y la comparación de los protocolos TCP y UDP en diferentes contextos de aplicación. Además, se explorarán las características de los paquetes UDP, como su estructura y los beneficios de su uso en comparación con TCP en ciertos escenarios de red.

**4.2**

**Paso 2 Verificar direcciones IP del cliente y el servidor**

Direccion IP Cliente

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

Dirección IP del servidor (Univirtual)

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

Analice los datos HTTP capturados. Utilice los resultados de captura de Wireshark para responder las siguientes preguntas:

**a. Complete la siguiente tabla con la información presentada en la sesión HTTP, hacer los mismo para el acceso a los tres servicios web:**

# Univirtual: https://univirtual.unicauca.edu.co/.

|  |  |
| --- | --- |
| Dirección IP del explorador Web  (cliente) | 192.168.128.37 |
| Dirección IP del servidor Web | 10.20.4.43 |
| Protocolo de la capa de transporte  (UDP/TCP) | TCP |
| Número de puerto del explorador Web | 56745 |
| Número de puerto del servidor Web | 443 |

# Recarga Biblioteca: http://biblio.unicauca.edu.co/.

|  |  |
| --- | --- |
| Dirección IP del explorador Web  (cliente) | 192.168.128.37 |
| Dirección IP del servidor Web | 10.20.5.11 |
| Protocolo de la capa de transporte  (UDP/TCP) | TCP |
| Número de puerto del explorador Web | 56749 |
| Número de puerto del servidor Web | 80 |

**3. Servidor http://ip\_del\_servidor/redes/public/login).**

|  |  |
| --- | --- |
| Dirección IP del explorador Web  (cliente) | 192.168.128.37 |
| Dirección IP del servidor Web | 192.168.128.25 |
| Protocolo de la capa de transporte  (UDP/TCP) | TCP |
| Número de puerto del explorador Web | 56753 |
| Número de puerto del servidor Web | 80 |

**b. ¿Qué computadora inició la sesión HTTP y cómo lo hizo?, responder la pregunta teniendo en cuenta la información teórica de los capítulos 1-4 del curso.**

Al revisar los datos del tráfico de red utilizando la aplicación Wireshark, lo primero que hicimos fue buscar los paquetes que incluían solicitudes HTTP. En este caso, tomamos como ejemplo la URL de la universidad, <http://biblio.unicauca.edu.co/> donde pudimos identificar la dirección IP de origen como 192.168.128.37.

Ahora se procede a realizar un análisis de la sesión, utilizando la información más detallada proporcionada por la aplicación, con el objetivo de comprender mejor su funcionamiento.Texto

Descripción generada automáticamente

En el análisis de la sesión HTTP y la comunicación subyacente, se observa que la conexión entre la computadora de origen y el servidor de destino se establece mediante el protocolo TCP (Protocolo de Control de Transmisión). Este protocolo es reconocido por su capacidad de brindar una comunicación confiable entre dos dispositivos en una red.

Durante la sesión, la computadora de origen inicia una solicitud de conexión utilizando el puerto de origen 56750, que corresponde a los puertos privados y dinámicos utilizados por la computadora para diversas aplicaciones. Esta solicitud se conoce como un "handshake" de tres vías, un proceso crucial en la inicialización de una conexión TCP. En este proceso, se establecen y confirman los números de secuencia iniciales para sincronizar ambas partes y prepararlas para la comunicación.

El proceso de tres vías implica lo siguiente:

* La computadora de origen envía un paquete SYN al servidor de destino para iniciar la conexión.
* El servidor de destino responde con un paquete SYN-ACK, confirmando la recepción del paquete SYN y estableciendo su propio número de secuencia.
* Finalmente, la computadora de origen envía un paquete ACK al servidor de destino, confirmando la recepción de la respuesta del servidor.

El puerto de destino, en este caso, es el número de puerto al que se envía la comunicación. Para HTTP, el puerto de destino es el 80, que es el puerto estándar utilizado para las solicitudes y respuestas de HTTP. Este hecho indica que la comunicación está dirigida específicamente a un servidor web que escucha el puerto 80 para procesar solicitudes HTTP.

**c. En TCP se establecen las conexiones usando el protocolo de acuerdo a tres vías (three-way handshake), utilice los resultados de las capturas para evidenciar este proceso. Explique en detalle cada vía en cada caso (soporte la explicación con capturas de pantalla de la vía correspondiente).**

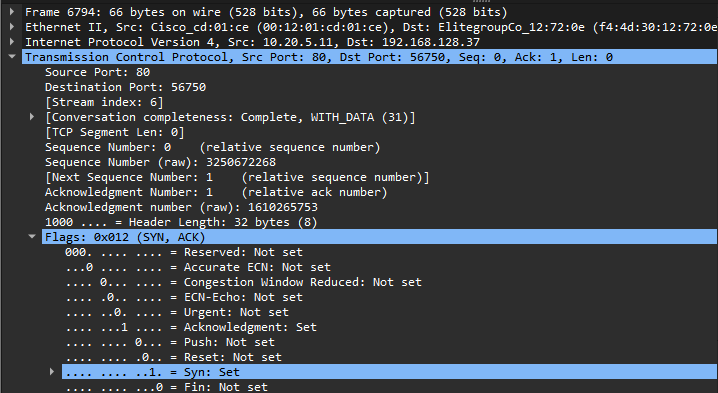
Ejemplo con biblio.unicauca.edu.co

**Paso 1: Solicitud de Conexión (SYN):**

Cuando el cliente desea establecer una conexión con el servidor, envía un paquete TCP especial conocido como "Segmento SYN". En este paquete, el cliente selecciona un número de secuencia inicial (ISN - Initial Sequence Number) para la comunicación y activa la bandera SYN estableciéndola en "1". Esta acción indica al servidor que el cliente está solicitando iniciar una conexión.Texto

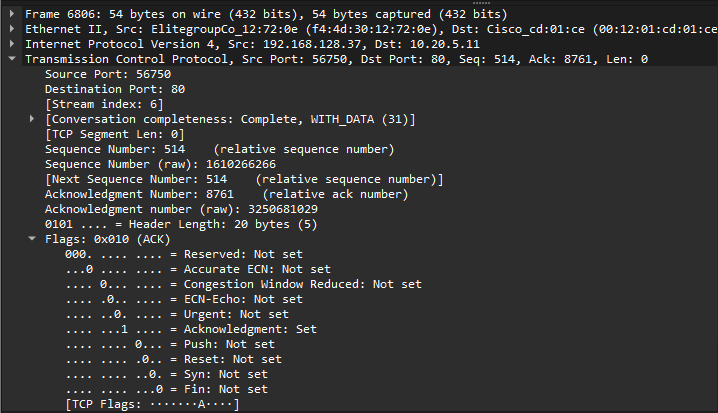
Descripción generada automáticamente

**Paso 2: Aceptación de Conexión (SYN-ACK):**

Cuando el servidor recibe el paquete SYN del cliente, responde con un paquete denominado "Segmento SYN-ACK", el cual indica que acepta la solicitud de conexión del cliente y está preparado para proceder con la comunicación.

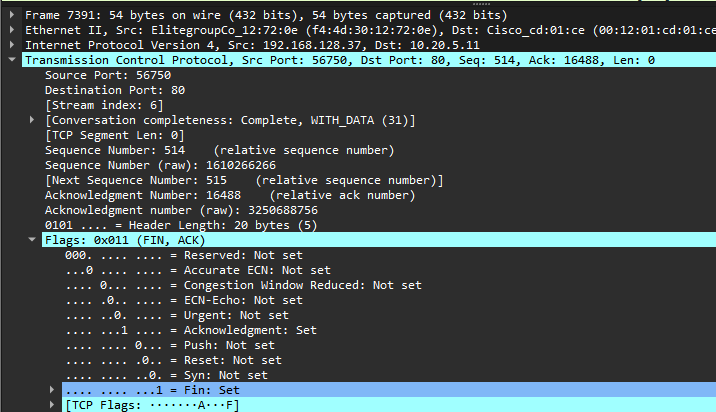
**Paso 3: Confirmación de Conexión (ACK):**

El cliente responde enviando un paquete denominado "Segmento ACK". En este paquete, el cliente incrementa nuevamente el número de secuencia y activa la bandera ACK, estableciéndola en "1". El número de ACK (Acknowledgment Number) se establece en el valor del número de secuencia del servidor incrementado en 1. Con este intercambio, tanto el cliente como el servidor han confirmado la conexión y están listos para iniciar la comunicación.

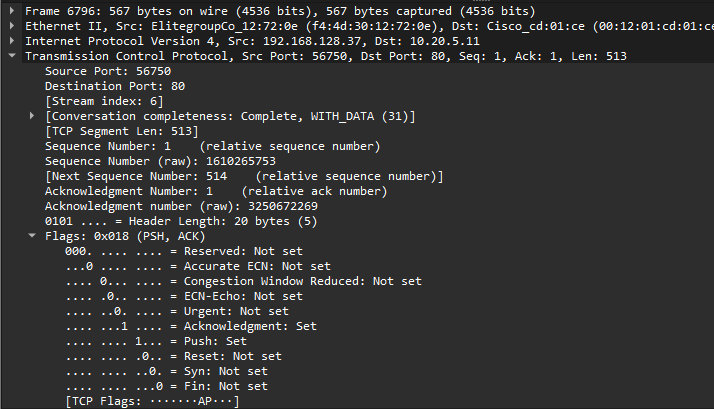


**d. ¿Qué computadora señaló inicialmente un fin a la sesión HTTP y cómo lo hizo?, explicar y dejar evidencia.**

En esta instancia, el remitente ha establecido las banderas ACK y FIN en 1, lo que señala que ha concluido la transmisión de datos. Al mismo tiempo, reconoce que ha recibido todos los datos enviados por la parte receptora.



1. **Resalte la primera línea del protocolo HTTP, una solicitud GET (Obtener) del explorador Web. Vaya a la segunda ventana de Wireshark (doble clic a la línea en mención) para examinar los protocolos en capas. Si es necesario, expanda los campos.**

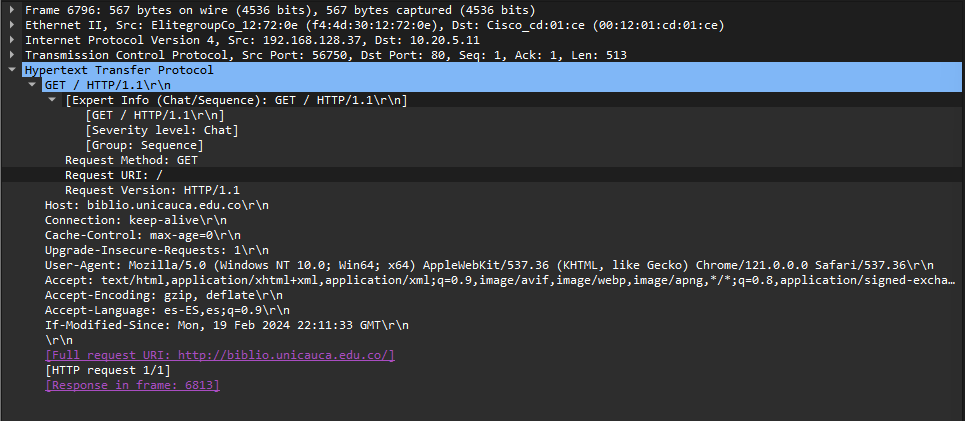


1. **¿Qué protocolo se lleva (encapsulado) dentro del segmento TCP?, dejar evidencia.** Protocolo HTTP

Texto

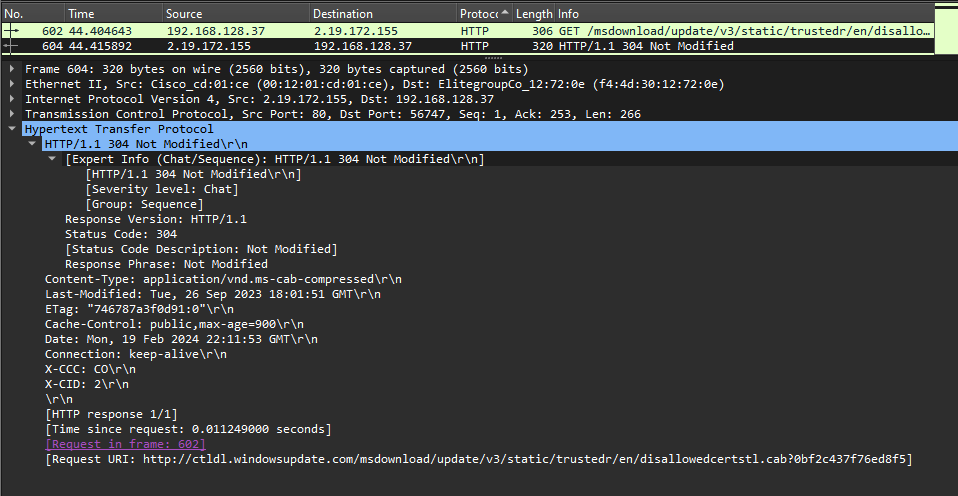
Descripción generada automáticamente

1. **Expanda el último registro de protocolo y cualquier subcampo. Ésta es la información real enviada al servidor Web. Complete la siguiente tabla utilizando la información del protocolo.**



|  |  |
| --- | --- |
| Versión del protocolo | HTTP/1.1 |
| Método de solicitud | GET |
| Solicitud URI | / |
| Idioma | Accept-Language: es-ES,es;q=0.9 |

1. **¿Cuál es la respuesta del servidor Web para la solicitud GET del cliente Web?, explicar y dejar evidencia.**

La respuesta HTTP/1.1 304 Not Modified es enviada por el servidor para comunicar al cliente que el recurso solicitado no ha sufrido modificaciones desde la última vez que fue accedido.

1. **¿Qué significa esta respuesta?**

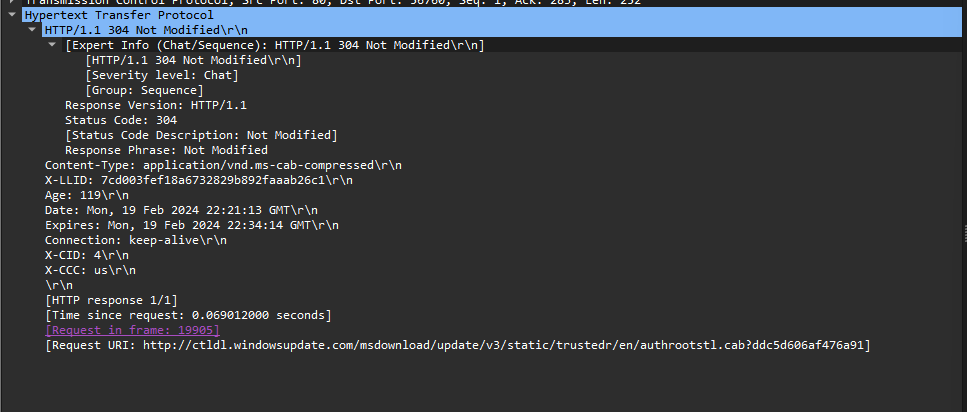
El código HTTP “304 Not Modified” indica que el recurso aún es cálido y se encuentra en caché en el cliente. Por lo que en lugar de enviar todo el contenido nuevamente, el servidor indica que simplemente puede volver a utilizar la versión en caché que ya posee, lo que ahora el ancho de banda y reduce la carga en el servidor de la red. Por lo que podemos entender que esta respuesta es parte de la funcionalidad de caché en HTTP, que permite a los navegadores y servidores ahorrar tiempo y recursos al evitar la descarga repetida de recursos que ni han cambiado.

1. **Desplácese hacia abajo de la ventana superior de Wireshark hasta que se muestre la segunda sesión de HTTP, actualizada. El significado de la acción de actualización se encuentra en la respuesta del servidor, 304 Not Modified (304**



**No modificado), localice esta línea en Wireshark.**

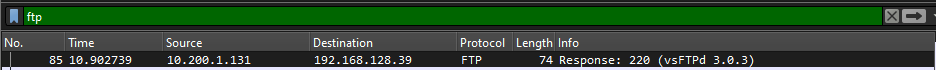
**Localizar esta información en Wireshark y dejar la respectiva evidencia.**



**4.3 Actividad 3. Captura y análisis de la comunicación FTP entre la computadora host del módulo y un servidor Web.**

**3. Vaya a la ventana de Wireshark del medio y expanda el protocolo FTP. FTP se comunica usando códigos, como HTTP. Responder:**

1. **¿Cuál es la respuesta 220 del servidor FTP?**



Texto

Descripción generada automáticamente

1. **Cuando el servidor FTP emitió una Respuesta: 331. Especifique la contraseña.**

**¿Cuál fue la respuesta del explorador Web?**

**331. especifique la contraseña**

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Respuesta del explorador web**



1. **¿Qué número de puerto utiliza el cliente FTP para conectarse al puerto 21 del servidor FTP?**

El puerto es el 63533.

Texto

Descripción generada automáticamente

1. **Cuando se transfieren datos, o con listados simples de directorios, se abre un nuevo puerto. Esto se llama modo de transferencia. El modo de transferencia puede ser activo o pasivo.**

**¿Cuál es el número de puerto de Datos FTP utilizado por el servidor FTP?**

El numero de puerto es el 29954.

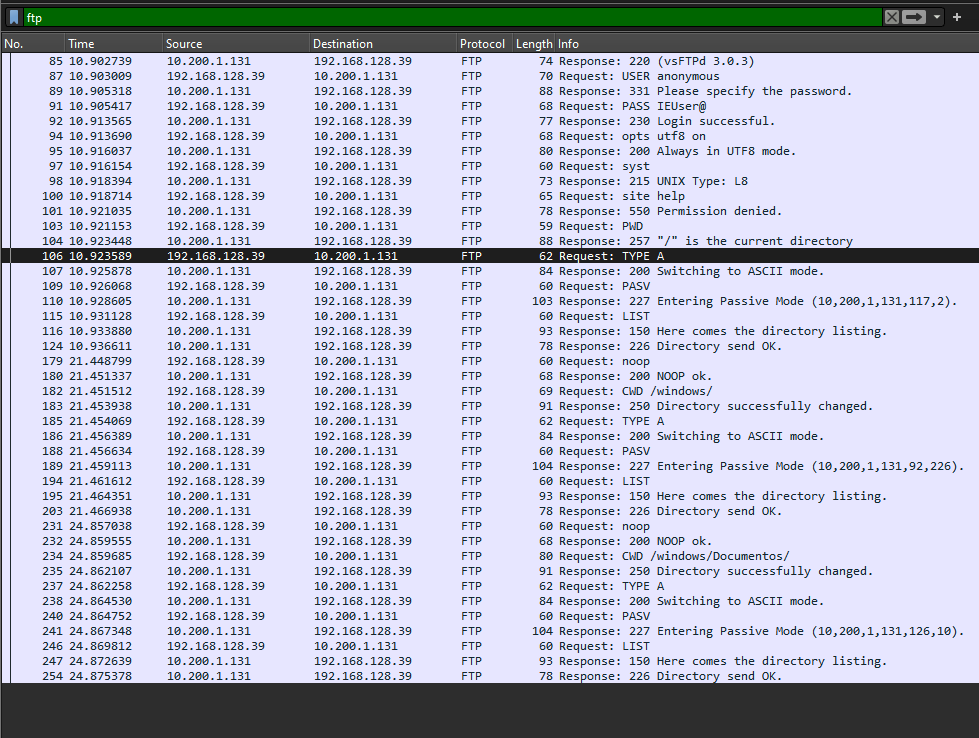
Texto

Descripción generada automáticamente

1. **Abra la captura de Wireshark FTP\_Command\_Line\_Client y observe la comunicación FTP. Cuál es la semejanza entre los clientes línea de comando y browser.**

La similitud encontrada en la recolección de tráfico de red entre el navegador y los clientes de línea de comandos radica en las respuestas estándar del servidor a los comandos enviados por ambos tipos de clientes. Tanto el navegador como los clientes de línea de comandos utilizan comandos FTP para realizar acciones específicas, como listar archivos, subir y descargar archivos, crear directorios, entre otros. Esto demuestra que los comandos FTP son similares independientemente de la interfaz utilizada.

Además, en ambos tipos de clientes se requiere autenticación para acceder al servidor FTP, utilizando credenciales como nombre de usuario y contraseña. En muchos casos, se utiliza "anonymous" como nombre de usuario y no se requiere contraseña, lo que permite interactuar con los archivos del servidor FTP, ya sea para descargarlos, subirlos, eliminarlos o renombrarlos. En resumen, tanto los clientes de línea de comandos como el navegador comparten similitudes en términos de cómo se comunican con los servidores FTP a través del protocolo FTP.**Tráfico de red del browser**

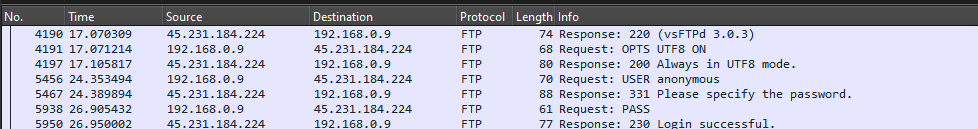


**4.4 Actividad 4. Reflexión**

**Explicar y evidenciar como los protocolos HTTP y FTP finalizan una sesión. Tráfico de red del cliente de línea de comandos.**

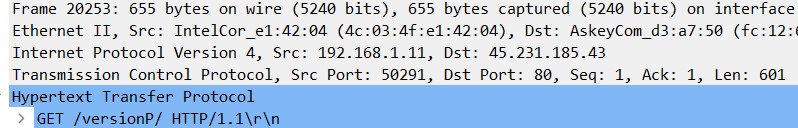
**FTP**

Inicialmente, el cliente solicita conectarse al servidor FTP a través del puerto 21. El servidor responde con el código 220, que indica "Servicio listo para un nuevo usuario". Luego, se inicia el proceso de autenticación de usuario y contraseña, lo que eventualmente conduce a un inicio de sesión exitoso.

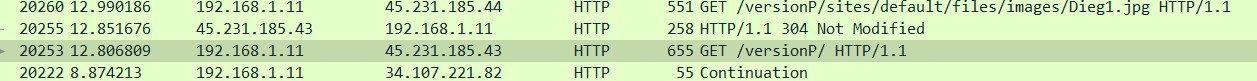


Para finalizar una sesión, el cliente envía una petición "QUIT" para cerrar la sesión de manera ordenada. Posteriormente, el servidor responde con el código "221", confirmando el cierre de la sesión. Este mensaje de respuesta suele incluir un mensaje de despedida o agradecimiento.

**HTTP**

Para iniciar sesión a través del protocolo HTTP, el cliente envía una solicitud al servidor a través del puerto 80. Una vez establecida la conexión, el cliente envía una solicitud HTTP al servidor utilizando métodos como GET, POST, etc. Estos métodos permiten al cliente solicitar recursos específicos o enviar datos al servidor para procesarlos.

En HTTP/1.1, luego de que el servidor envía la respuesta es posible cerrar la sesión.



1. **Amplíe el protocolo de datagramas de usuario en el panel de detalles del paquete haciendo clic en el signo más (+). Observe que hay solo cuatro campos. Identifique y explique cada uno de dichos campos. Dejar evidencia del resultado obtenido.**

Texto

Descripción generada automáticamente

**Source Port:** Número de puerto de origen

**Destination**: Indica el número de puerto del destinatario

**Length:** Longitud total del paquete UDP en bytes

**Checksum:** Se utiliza para verificar la integridad de los datos en el paquete UDP

1. **¿Cuál es la longitud del datagrama (bytes)? ¿Cuántos bytes pertenecen a la cabecera y cuantos pertenecen a la consulta DNS?**

Con el UDP payload, podemos determinar cómo se distribuyen los bytes entre la cabecera UDP y la consulta DNS. Dado que la cabecera UDP tiene una longitud fija de 8 bytes, los 32 bytes totales del payload UDP se distribuyen entre la cabecera UDP y la consulta DNS.

Después de restar los 8 bytes de la cabecera UDP del total de 32 bytes del payload, quedan 24 bytes disponibles para la carga útil o la consulta DNS. Por lo tanto, en este caso, tendríamos 24 bytes para la consulta DNS.Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**En Wireshark, cuando ves la información "UDP playLoad 32 bytes", significa que el tamaño del contenido del datagrama UDP es de 32 bytes. Sin embargo, debes tener en cuenta que este tamaño incluye tanto la cabecera UDP como los datos reales que se están transmitiendo. La cabecera UDP en sí misma ocupa 8 bytes.**

**La cabecera UDP consta de 8 bytes en total y se divide en los siguientes campos:**

El análisis del datagrama UDP revela que consta de varios campos:

* Puerto de origen (2 bytes): Indica el puerto del remitente.
* Puerto de destino (2 bytes): Indica el puerto del receptor.
* Longitud (2 bytes): Indica el tamaño total del datagrama UDP, incluida la cabecera y los datos.
* Checksum (2 bytes): Se utiliza para verificar la integridad de los datos, aunque su cálculo no es obligatorio.

Dado que el datagrama UDP tiene una longitud total de 32 bytes y la cabecera UDP ocupa 8 bytes, podemos deducir que los datos reales (carga útil) ocupan 32 - 8 = 24 bytes.

**8. Observe el datagrama de respuesta (DNS→Cliente) y responda:**

**Observe la cantidad de bytes y compárelo con el tamaño del datagrama de la consulta, cual es más grande? Dejar evidencia**

Podemos evidenciar que la respuesta tiene un tamaño mayor:

Texto

Descripción generada automáticamente

con un tamaño de 48 bytes.

**¿Cuál fue el comportamiento de los números de puerto? Observe que hay solo cuatro campos. Identifique y explique cada uno de dichos campos. Dejar evidencia del resultado obtenido.**

Si el cliente desea comunicarse con el servidor, envía un datagrama UDP al puerto de destino del servidor “53”. El servidor recibe el datagrama y procesa la solicitud.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

El servidor envía una respuesta al cliente mediante un datagrama UDP dirigido al puerto de origen del cliente, que en este caso es "61683". Este puerto de origen es seleccionado de manera aleatoria y dinámica por el sistema operativo del cliente para recibir la respuesta del servidor.Texto

Descripción generada automáticamente

* **Source Port (Puerto de Origen):** Es el número de puerto del remitente del paquete UDP.
* **Destination Port (Puerto de Destino):** Indica el número de puerto del destinatario al que se enviará el paquete UDP.
* **Length (Longitud):** Representa la longitud total del paquete UDP en bytes, incluyendo la cabecera y los datos.
* **Checksum (Suma de Verificación):** Se utiliza para verificar la integridad de los datos en el paquete UDP y detectar posibles errores en la transmisión.

**9. ¿Cuáles son los beneficios de utilizar UDP en lugar de TCP como protocolo de transporte para servicios o protocolos como DNS o DHCP o por ejemplo streaming de audio y video?**

Utilizar UDP en lugar de TCP para servicios como DNS, DHCP o streaming de audio y video ofrece varios beneficios significativos. UDP, al ser un protocolo de transporte liviano y sin conexión, proporciona una menor sobrecarga de procesamiento y menor latencia en comparación con TCP. Esto es especialmente beneficioso para servicios en tiempo real, como el streaming de audio y video, donde la prioridad es la velocidad y la entrega rápida de datos sobre la confiabilidad de la entrega. Además, UDP es ideal para servicios donde la pérdida ocasional de paquetes no es crítica, como en DNS o DHCP, ya que estos protocolos pueden tolerar la pérdida de algunos paquetes sin afectar significativamente la funcionalidad. En resumen, UDP ofrece una alternativa eficiente y de bajo nivel para servicios que requieren comunicaciones rápidas y en tiempo real, sacrificando la confiabilidad en favor de un rendimiento más ágil y una menor sobrecarga de red

**10. ¿Cuáles son los beneficios de utilizar TCP en lugar de UDP como protocolo de transporte para protocolos de mail (POP3, SMTP), conexiones remotas (Telnet, SSH) y/o transferencia de archivos (FTP)?**

Utilizar TCP en lugar de UDP para protocolos de mail (POP3, SMTP), conexiones remotas (Telnet, SSH) y transferencia de archivos (FTP) ofrece varios beneficios clave. TCP, al ser un protocolo orientado a la conexión y confiable, garantiza la entrega ordenada y sin errores de los datos, lo que es esencial para aplicaciones que requieren integridad de los datos, como el correo electrónico y la transferencia de archivos. Además, TCP gestiona automáticamente la retransmisión de paquetes perdidos y el control de congestión, lo que garantiza una entrega confiable incluso en redes propensas a la pérdida de paquetes o congestión. Esto resulta especialmente crítico para protocolos como SMTP y POP3, donde la pérdida de correos electrónicos o la corrupción de datos pueden ser catastróficas. Además, para conexiones remotas a través de Telnet y SSH, la confiabilidad y la integridad de los datos son fundamentales para garantizar una comunicación segura y sin interrupciones. En resumen, TCP proporciona los mecanismos necesarios para garantizar la entrega confiable y ordenada de los datos, lo que lo convierte en la elección ideal para aplicaciones donde la integridad de los datos es primordial.

Referencias

[1] Cisco Networking Academy. (s.f.). CCNA 1: Introducción a las redes.

[2]http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro27/47\_definicin\_de\_proto colos\_de\_conexin\_y\_sin.html

Url del repositorio:

<https://github.com/JhossefCons/REDES>

Url del video:

https://youtu.be/XZgZmkdu5Sk