Tshark

Tshark es la herramienta de línea de comandos asociada a Wireshark y es excelente para auditar y analizar tráfico de red.

Guía de Ejercicios de Tshark

Para realizar estos ejercicios, necesitarás **tshark** (instalado con sudo apt install tshark) y **tcpdump** (generalmente ya instalado).

sudo apt install tshark

Preparación: Crear un Archivo de Captura Base

Antes de cualquier ejercicio, crea un archivo de captura (sample.pcap) que contenga una mezcla de tráfico HTTP, DNS y SSH.

1. **Captura de ejemplo (en una terminal):**

   sudo tcpdump -i <interfaz> -c 300 -w sample.pcap
   sudo tcpdump -i enp0s3 -c 300 -w sample.pcap

   Reemplaza <interfaz> por la tuya (ej. eth0 o enp0s3). La opción -c 300 limita la captura a 300 paquetes.

2. Generar Tráfico (en otra terminal, mientras se captura):

   Haz un ping a un sitio, una resolución DNS y una solicitud HTTP.

   ping -c 3 google.com
   host google.com
   wget -qO- www.example.com > /dev/null

3. **Detener la Captura:** Vuelva a la primera terminal y detenga la captura (si no terminó automáticamente) con **Ctrl+C**.

Ejercicio 1: Resumen Básico de la Captura

Este ejercicio muestra información clave de los paquetes en un formato similar a la vista principal de Wireshark.

Objetivo	Comando tshark	Descripción
Mostrar los primeros 10 paquetes	`tshark -r sample.pcap -c 10`	Muestra el número, tiempo, IP de origen/destino, protocolo e información del paquete.
Contar Paquetes	`tshark -r sample.pcap | wc -l`	número que indica cuántos paquetes se grabaron en el archivo
Mostrar sólo las IPs	`tshark -r sample.pcap -T fields -e ip.src -e ip.dst -c 5`	Muestra solo los campos de IP de origen y destino para los primeros 5 paquetes.

Ejercicio 2: Filtrado por Protocolo (Display Filters)

Los filtros de visualización de Tshark son los mismos que se usan en Wireshark. Son esenciales para el análisis.

Objetivo	Archivo de Ejemplo	Comando tshark
Mostrar sólo tráfico DNS	Generado en el paso de Preparación.	`tshark -r sample.pcap -Y "dns"`
Mostrar sólo peticiones HTTP	Generado en el paso de Preparación.	`tshark -r sample.pcap -Y "http.request"`
Tráfico entre dos IPs	Genere tráfico entre su VM y la IP del *host* (ej. 192.168.1.1).	`tshark -r sample.pcap -Y "ip.addr == 192.168.1.1"`

Objetivo	Archivo de Ejemplo	Comando tshark
Tráfico en un puerto específico	Genere tráfico SSH/Glances.	tshark -r sample.pcap -Y "tcp.port == 22"

Ejercicio 3: Extracción de Datos Específicos (Campos)

Utiliza la opción **-T fields** para extraer valores concretos de los paquetes, útil para *scripting* o auditorías.

Objetivo	Comando de Ejemplo	Campos a Extraer (-e)
Nombres de Host Solicitados (DNS)	tshark -r sample.pcap -Y "dns" -T fields -e dns.qry.name	Extrae el campo dns.qry.name.
Códigos de Estado HTTP	Genere tráfico HTTP con errores y éxito (ej. wget a una página que no existe).	tshark -r sample.pcap -Y "http.response" -T fields -e http.response.code
Tiempo de Respuesta (TCP)	Con el archivo sample.pcap.	tshark -r sample.pcap -T fields -e tcp.time_delta

Ejercicio 4: Filtros de Captura (Capture Filters)

A diferencia de los filtros de visualización (-Y), los **filtros de captura** (-f) filtran los paquetes *antes* de que se guarden en el archivo, ahorrando espacio y recursos. Utilizan la sintaxis de **tcpdump (BPF)**.

Objetivo	Creación del Archivo de Ejemplo	Comando tshark
Capturar sólo paquetes ICMP	**Paso 1:** sudo tcpdump -i enp0s3 -c 10 -w icmp.pcap 'icmp' **Paso 2:** ping -c 10 google.com	tshark -r icmp.pcap -Y "icmp" (Verificará que solo hay ICMP)
Capturar solo tráfico de red (no local)	**Paso 1:** sudo tcpdump -i enp0s3 -w non_local.pcap 'not port 22'	tshark -r non_local.pcap -Y "tcp.port == 22"

Que la salida de tshark esté vacía es la **confirmación** de que los paquetes del puerto 22 (SSH) fueron excluidos de la captura.

tshark -r non_local.pcap

Comando	**sudo tcpdump -i enp0s3 -w non_local.pcap 'not port 22'**
Filtro Aplicado	**'not port 22'**: Este filtro instruye a tcpdump a capturar y guardar **todos** los paquetes, excepto aquellos cuyo puerto de origen o destino sea el puerto 22 (SSH).
Resultado	169 packets captured
Conclusión	tcpdump capturó 169 paquetes que **no** contenían tráfico SSH y los guardó en non_local.pcap.

2. Verificación de Exclusión Exitosa (tshark)

Comando	**tshark -r non_local.pcap -Y "tcp.port == 22"**
Filtro Aplicado	**"tcp.port == 22"**: Este filtro de visualización intenta mostrar **solo** los paquetes TCP que usen el puerto 22.
Resultado	*(Salida vacía)*
Conclusión	Si tshark no muestra nada, significa que **no encontró ningún paquete** que cumpliera con la condición de tener el puerto 22. Esto verifica que el filtro de exclusión de tcpdump ('not port 22') funcionó correctamente al momento de la captura.

Ejercicio 5: Reconstrucción de Streams TCP

Este ejercicio es fundamental para la seguridad y la auditoría, ya que permite ver la conversación completa entre dos puntos.

Objetivo	**Creación del Archivo de Ejemplo**	**Comando tshark**
Ver la conversación HTTP	Genera tráfico HTTP simple (ej. wget www.example.com).	tshark -r sample1.pcap -z "follow,tcp,ascii,0"
Ver sólo la información de un Stream	*El número 0 se refiere al primer stream TCP detectado en el archivo.*	tshark -r sample1.pcap -z "follow,tcp,ascii,1"

Nota sobre follow,tcp: El número al final (0 o 1) indica el índice del *stream* TCP. Si omite el número, Tshark le mostrará un resumen de los *streams* disponibles.

Necesitarás dos terminales abiertas en tu máquina Ubuntu para este ejercicio.

1. Paso Preliminar: Limpiar Archivos de Captura

Antes de comenzar, asegúrate de que no haya un archivo sample1.pcap antiguo:

Elimina el archivo de captura anterior si existe

rm -f sample1.pcap

2. Paso de Captura: Crear el Archivo de Ejemplo

En la **Terminal 1**, vamos a capturar el tráfico que necesitamos. Usaremos el *flag* -c 50 para limitar la captura a 50 paquetes y asegurar que se detenga rápidamente. También aplicaremos un filtro port 80 or port 443 para centrarnos en el tráfico web.

sudo tcpdump -i enp0s3 -c 50 -w sample1.pcap 'port 80 or port 443'

La Terminal 1 mostrará: tcpdump: listening on enp0s3, ...

3. Paso de Generación: Crear Tráfico Web

En la **Terminal 2**, ejecuta los siguientes comandos para generar una solicitud web que será capturada en la Terminal 1. Usaremos **wget** para garantizar una solicitud HTTP/HTTPS completa:

Genera una solicitud HTTP/HTTPS y descarta la salida

```
wget -qO- www.example.com > /dev/null
```

Si quieres que tcpdump llegue a los 50 paquetes con el filtro port 80 or port 443, simplemente genera más solicitudes web.

En la **Terminal 2**, repite el comando wget varias veces hasta que veas que la Terminal 1 se detiene:

El comando tcpdump en la Terminal 1 debería detenerse automáticamente tras capturar 50 paquetes o poco después de que wget termine.

4. Paso de Análisis (Parte A): Ver el Resumen de Streams

Ahora que tienes el archivo sample1.pcap, puedes usar tshark para analizar y reconstruir. Primero, pide a tshark que te muestre los *streams* TCP disponibles **sin especificar un número de *stream***:

```
tshark -r sample1.pcap -z "follow,tcp,ascii,0"
```

- **Resultado Esperado:** tshark te devolverá una tabla (o un listado) que te dirá algo como: Stream 0: <IP> -> <IP>, Stream 1: <IP> -> <IP>, etc. Esto te da el **índice numérico** de los *streams*.

5. Paso de Análisis (Parte B): Reconstruir el Primer Stream

Basándote en el resumen anterior (o asumiendo que el *stream* 0 o 1 contiene el tráfico HTTP/HTTPS de wget), usa el comando para ver la conversación completa.

Ver el Stream 0 (Conversación Completa)

```
tshark -r sample1.pcap -z "follow,tcp,ascii,0"
```

- **Resultado Esperado:** Verás el texto sin procesar de la solicitud HTTP (GET /...) y la respuesta del servidor (HTTP/1.1 200 OK...), lo que demuestra la reconstrucción de la conversación entre tu cliente y el servidor web.

Ver el Stream 1 (Si es Necesario)

Si el *Stream 0* no era el tráfico web, puedes probar el siguiente índice:

```
tshark -r sample1.pcap -z "follow,tcp,ascii,1"
```

Este ejercicio confirma que puedes tomar un archivo de captura y reconstruir la secuencia exacta de paquetes TCP para ver el contenido completo de la conversación.

Ejercicio 6: Estadísticas de Red (Conversaciones)

Utilice la opción **-z** de Tshark para generar estadísticas útiles, como un resumen de las conversaciones (pares de direcciones IP).

Objetivo	**Comando tshark**	**Descripción**
Listar las conversaciones IPv4	`tshark -r sample.pcap -z conv,ip`	Muestra qué IPs se comunicaron, el número de paquetes y bytes transferidos.
Estadísticas por Protocolo	`tshark -r sample.pcap -z io,phs`	Muestra el porcentaje de paquetes de cada protocolo (Ethernet, IP, TCP, etc.).

Objetivo	Comando tshark	Descripción
Gráfico de E/S por segundo	tshark -r sample1.pcap -z io,stat,1	Esto te muestra la cantidad de actividad de red en el archivo de captura desglosada por intervalos de un segundo..

TShark y Wireshark

TShark es la versión de línea de comandos (CLI) de Wireshark.

Ambas herramientas comparten el mismo motor principal de análisis de protocolos, pero se diferencian en la forma en que se interactúa con ellas.

1. TShark: La Herramienta de Consola

TShark está diseñado para la **automatización** y el **uso sin interfaz gráfica** (headless).

Característica	Descripción
Interfaz	Línea de comandos (CLI).
Uso Típico	Captura de tráfico en servidores remotos sin GUI, *scripts* automatizados, filtros complejos, análisis de grandes archivos .pcap en lote.
Rendimiento	Suele ser más eficiente en el uso de recursos, ya que no necesita renderizar gráficos.
Salida	Muestra los resultados en texto plano en la terminal.

TShark es esencial cuando: necesitas capturar tráfico en un servidor Linux/Unix al que accedes por SSH, o cuando quieres automatizar el filtrado de paquetes dentro de un *script* de Bash.

2. Wireshark: La Herramienta Gráfica

Wireshark es la aplicación de escritorio que la mayoría de la gente conoce, proporcionando una interfaz de usuario completa.

Característica	Descripción
Interfaz	Interfaz Gráfica de Usuario (GUI).
Uso Típico	Análisis interactivo, inspección visual de paquetes, seguimiento de flujos TCP, identificación rápida de anomalías, uso de gráficos estadísticos.
Rendimiento	Requiere más recursos del sistema (CPU y memoria) para renderizar la GUI.
Salida	Presentación visual y organizada de los datos en paneles de listado, detalle y *raw data*.

Resumen

Aspecto	TShark	Wireshark
Motor de Análisis	Comparten el mismo motor.	Comparten el mismo motor.
Filtros	Utiliza los **mismos filtros de visualización** (display filters).	Utiliza los **mismos filtros de visualización**.
Archivos	Puede leer y escribir los mismos archivos de captura (.pcap, .pcapng).	Puede leer y escribir los mismos archivos de captura.
Diferencia Clave	**Automatización y Consola.**	**Visualización e Interacción.**

En esencia, **TShark te permite hacer todo lo que Wireshark puede hacer a nivel de análisis de paquetes y filtrado, pero sin necesidad de una pantalla gráfica.**

Pasos para Instalar Wireshark en Ubuntu 24.04

Abre tu terminal de Ubuntu y ejecuta los siguientes comandos:

1. Actualizar el Índice de Paquetes

Asegúrate de que tu lista de paquetes locales esté actualizada:

```
sudo apt update
```

2. Instalar Wireshark

Instala el paquete wireshark desde los repositorios.

```
sudo apt install wireshark -y
```

3. Configurar Permisos (¡Paso Importante!)

Durante la instalación, se te preguntará si deseas permitir que **usuarios no *root*** (usuarios normales, como tu usuario feval) puedan capturar paquetes.

- **Selecciona Sí o Yes** (la opción recomendada). Esto es crucial para que no tengas que usar sudo cada vez que quieras ejecutar Wireshark, lo cual es una práctica de seguridad más segura.

Si elegiste Sí, debes añadir tu usuario al grupo wireshark para aplicar el permiso:

```
sudo usermod -aG wireshark $USER
```

El $USER es una variable que toma automáticamente el nombre de tu usuario actual (feval en tu caso).

4. Aplicar Cambios

Para que los cambios de grupo surtan efecto (el permiso para capturar tráfico), debes cerrar tu sesión de terminal e iniciar una nueva, o **reiniciar la máquina virtual.**

Después de eso, podrás iniciar Wireshark escribiendo wireshark en la terminal o buscándolo en el menú de aplicaciones de Ubuntu.

```
wireshark
```