Ingeniería en Sistemas Computacionales

Centro de Ciencias Básicas

5º C

Proyecto redes I

Alumnos:

353079 Tadeo Andrade Sustaita

350114 Cindy Fabiola Hernández Muñoz

350944 Reyli Uvaldo Martínez Hernández

281547 Juan Carlos Uriarte Padilla

Profesor: Javier Santiago Cortes López

Fecha de entrega: 17 de diciembre de 2024

Contenido

REPORTE DE PROYECTO: PACKET SNIFFER

Introducción

El objetivo principal de este proyecto es el desarrollo de un software que implemente las funciones esenciales de un Packet Sniffer, permitiendo la captura y el análisis detallado de paquetes de red. Para ello, se utilizará el lenguaje de programación C y la biblioteca libpcap, que proporciona acceso a la información del tráfico de red. La herramienta está diseñada para ser intuitiva, eficiente y fácil de usar, ofreciendo a los usuarios la capacidad de visualizar y exportar el tráfico de red de manera efectiva, facilitando así tareas de monitoreo y diagnóstico de redes.

Objetivos

El proyecto busca crear un software que:

- Capture paquetes de red en tiempo real.
- Permita aplicar filtros para analizar el tráfico.
- Tenga una interfaz fácil de usar.
- Exporte los resultados a un archivo CSV.

Requerimientos técnicos

Para el desarrollo de este proyecto, decidimos utilizar el lenguaje de programación C, ya que todos los integrantes del equipo tenemos conocimientos previos en este lenguaje, lo que facilita su implementación y nos permite desarrollar el software de manera más eficiente. Además, optamos por trabajar en Linux, ya que a cada uno de nosotros nos resultó más fácil realizar la primera parte del proyecto en este sistema operativo, aprovechando su compatibilidad y las herramientas disponibles, como libpcap, que es esencial para la captura de paquetes de red.

Desarrollo y Funcionalidad

- Captura de paquetes: Se implementó la funcionalidad básica para capturar paquetes de red en tiempo real utilizando la librería libpcap. Esta herramienta permite acceder al tráfico de red y monitorear la información que fluye a través de la red.
- Filtros de captura: El software permite aplicar diversos filtros de captura, como IP de destino, puerto de origen, puerto de destino y protocolo (TCP, UDP, ICMP, entre otros), lo que facilita el análisis de tráfico específico y la gestión de grandes volúmenes de datos.

- Interfaz de usuario: La interfaz de usuario fue diseñada para ser amigable y fácil de usar. Se dividió en áreas claramente definidas, que incluyen:
    - Área de tráfico capturado: Muestra los paquetes de red que se están capturando en tiempo real.
    - Área de contenido "raw": Muestra el contenido crudo del paquete seleccionado, permitiendo un análisis detallado del mismo.
- Exportación de datos: El tráfico capturado se puede exportar fácilmente a un archivo CSV, lo que permite su posterior análisis y almacenamiento en otros programas, como hojas de cálculo.

Manual del usuario

Se elaboró un manual de usuario en formato electrónico, que explica detalladamente cómo utilizar el software, aplicar filtros y exportar los resultados.
El manual incluye los siguientes apartados:

- Introducción al uso del software.
- Cómo iniciar y detener la captura de paquetes.
- Aplicación de filtros.
- Exportación de datos.

Futuras mejoras

Para seguir mejorando el software, se sugerimos las siguientes acciones:

- Interfaz gráfica más completa: Desarrollar una interfaz gráfica más avanzada que facilite una experiencia de usuario más intuitiva y visual, permitiendo a los usuarios interactuar de manera más eficiente con las diferentes funcionalidades del programa.
- Opciones de filtrado más avanzadas: Añadir más opciones de filtrado para un análisis más avanzado.
- Captura de tráfico en redes más complejas: Incluir la capacidad de capturar tráfico en redes más complejas.

Conclusiones

Tadeo: Este proyecto fue un éxito en muchos aspectos, ya que logramos cumplir con los objetivos propuestos. Desarrollamos una herramienta funcional para la captura y análisis de paquetes de red, que facilita el acceso eficiente a los datos de tráfico utilizando la librería libpcap. Las opciones de exportación añadidas también mejoraron la utilidad del software,

permitiendo un análisis más profundo y organizado del tráfico capturado. Me siento satisfecho con los resultados obtenidos y con las lecciones aprendidas durante el desarrollo de este proyecto.

Fabiola: Aunque no me siento completamente satisfecha con el resultado final, considero que logramos cumplir con algunos de los objetivos planteados al inicio del proyecto. Se desarrolló una herramienta funcional para la captura y análisis de paquetes de red, con una interfaz amigable y opciones útiles para los usuarios. A pesar de los desafíos, me quedo con valiosos aprendizajes y lecciones que me ayudarán en futuros proyectos. Espero poder continuar mejorando esta herramienta y completarla en una siguiente oportunidad.

Reyli: A lo largo de este proyecto, se logró desarrollar una herramienta funcional para la captura y análisis de paquetes de red, cumpliendo con los objetivos propuestos al inicio. El uso de la librería libpcap fue fundamental para acceder de manera eficiente a los datos de red. Además, la capacidad de exportar los resultados a archivos añadió una funcionalidad valiosa al software. Aunque siempre hay áreas de mejora, estoy satisfecho con el progreso alcanzado y con lo aprendido durante el desarrollo del proyecto.

Juan Carlos: El proyecto fue una buena oportunidad para aprender y aplicar lo que sabíamos. Logramos crear una herramienta funcional para capturar y analizar paquetes de red con libpcap, y la opción de exportar datos le dio más utilidad al software. Aunque hubo algunos retos, me siento satisfecho con lo que se logró y con lo que se aprendió en el proceso.

Bibliografías

Wireshark · Go Deep. (s. f.). Wireshark. https://www.wireshark.org/

Tanwar, P. (2021, 16 julio). Capturing packets in C program using LIBPCaP | Open Source. *Open Source For You*. https://www.opensourceforu.com/2011/02/capturing-packets-c-program-libpcap/

Talal. (2023b, agosto 2). Building a packet sniffer: the basics | Medium. *Medium*. https://talalio.medium.com/building-a-packet-sniffer-9460f394041

Talal. (2024, 26 abril). Building a packet sniffer Part 2 : Layers headers | Medium. *Medium*. https://talalio.medium.com/building-a-packet-sniffer-part-2-6fb33bd68d53

Talal. (2023, 2 agosto). Building a packet sniffer Part 3: Filters | Medium. *Medium*. https://talalio.medium.com/building-a-packet-sniffer-part-3-a404e60d91c5