Universidad Autónoma de Aguascalientes.

Centro de Ciencias Básicas.

“Documentación Proyecto Redes del Analizador de paquetes”

Materia: Redes de Computadoras I

Profesor: Javier Santiago Cortez López

Integrantes del equipo:

ID 349107 Emilio Ortiz Romo

ID 281466 Ricardo Almada Diaz

ID 349964 Ilse Jacqueline Martínez Espinosa

ID 349397 Uriel Rodríguez Guadarrama

Introducción:

Para la actualidad, prácticamente todos los individuos cuentan con dispositivos conectados a una red inalámbrica, por lo que el análisis de las redes inalámbricas y el trafico concurrente en las mismas están siendo ampliamente estudiados e investigados con fines tanto de seguridad como de estadística e incluso estabilidad y funcionamiento de un sistema. Para este tipo de intervenciones se usan herramientas como los packet sniffer, software que consiste en analizar los datos que se transmiten en una red inalámbrica. Permite observar el tráfico en tiempo real, descomponiendo el contenido de cada paquete para mostrar su contenido. Estos pueden contener datos técnicos de mucha relevancia según el tipo de búsqueda que se este haciendo. En este proyecto se tratará la elaboración de un packet sniffer desglosando el desarrollo y funcionamiento de este.

Objetivo: El proyecto tiene como objetivo desarrollar un packet sniffer utilizando las librerías npcap y libcap para Windows y Linux respectivamente, aportándole a la estructura herramientas de interacción para el usuario final.

Marco Teórico:

1.1 Antecedentes.

Primeros packet sniffers

En los años 1980, herramientas como tcpdump (desarrollada para UNIX en 1988 por Van Jacobson y Craig Leres) se convirtieron en pioneras en la captura y análisis de paquetes. Tcpdump permitió a los administradores observar el tráfico de red en un formato legible, usando la biblioteca libpcap.

Años 1990: Comercialización y herramientas avanzadas

A medida que las redes crecieron en complejidad, aparecieron herramientas más sofisticadas como Wireshark (antes Ethereal), lanzada en 1998. Estas ofrecían interfaces gráficas y análisis detallado de protocolos, democratizando el acceso a la tecnología de sniffing. También se popularizaron aplicaciones maliciosas que usaban sniffing para capturar contraseñas y datos sensibles en redes mal protegidas.

Años 2000: Seguridad en redes y sniffers especializados

Con el auge de Internet, los sniffers comenzaron a ser fundamentales para el diagnóstico de redes empresariales, la detección de intrusiones y el análisis forense. Se empezaron a implementar contramedidas como el cifrado (HTTPS, SSH) y redes segmentadas para dificultar la captura no autorizada de datos.

Actualidad: Packet sniffers modernos

Los sniffers actuales, como Wireshark, Nmap o Tshark, son herramientas avanzadas que soportan una amplia variedad de protocolos y permiten análisis en tiempo real. Su uso va desde la administración legítima de redes hasta la investigación de ciberseguridad y pruebas de penetración.

1.2 Herramientas utilizadas.

Fue utilizado npcap para el desarrollo en Windows del proyecto, manteniéndose en el entorno de desarrollo de Visual Studio para el mantenimiento de una terminal apta para la compilación. Se implementaron librerías adicionales para el uso de interfaces y para la deconstrucción e interpretado del contenido de los paquetes. Para el uso de gráficos e interfaces se utilizó la librería Windows.h, la cual da soporte de forma nativa a ventanas y gráficos interpretables en un sistema como Windows, mientras que se utilizó la librería vector para el guardado e interpretación de la estructura y contenido de los paquetes.

1.3 Funcionalidad del proyecto.

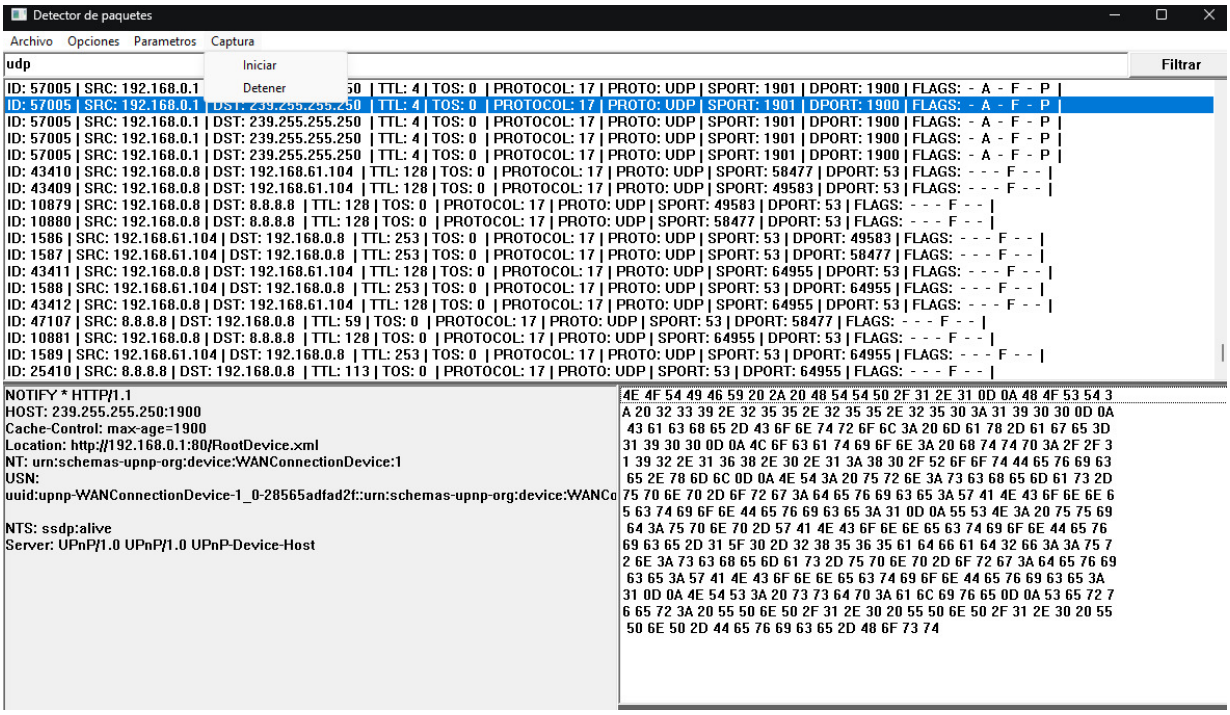
El proyecto permite al usuario, tras haber elegido el adaptador de red desde el cual se va a realizar la escucha, ver todos los paquetes que hay en el trafico de red a la cual esté conectada el mismo, permitiendo filtros y búsquedas más especificas según las necesidades del usuario. .El programa dentro de sus capacidades permite la exportación de una escucha de paquetes a un archivo de tipo .csv, de manera que la escucha de paquetes puede ser analizada en una archivo externo acorde a las necesidades del usuario.

Desarrollo:

Para la construcción del proyecto fue utilizada la base de un packet sniffer desarrollado con la librería libcap para Linux, fue necesario implementarlo en su contraparte para Windows, requiriendo algunas adaptaciones desde los comandos, funciones utilizadas y tipo de estructuras a manejar dentro del código.

Para el contenido del proyecto, se muestra una pantalla con 5 secciones, la pantalla principal de búsqueda, que es donde se muestran todos los paquetes escuchados, las dos pantallas inferiores donde se muestran los contenidos en crudo y en desempaquetado del paquete seleccionado en la pantalla principal. Finalmente se cuenta con una barra superior en la que se permite seleccionar el dispositivo de red desde el cual se realizara la escucha de paquetes, seleccionar los elementos de

filtrado, ya sea por tipo de paquete, protocolo, IP de envío, dirección o algún otro elemento de la preferencia del usuario. (ver imagen 1)



```
udp
```

```
ID: 57005 | SRC: 192.168.0.1                     50  | TTL: 4 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 1901 | DPORT: 1900 | FLAGS:  - A - F - P |
ID: 57005 | SRC: 192.168.0.1 | DST: 239.255.255.250  | TTL: 4 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 1901 | DPORT: 1900 | FLAGS:  - A - F - P |
ID: 57005 | SRC: 192.168.0.1 | DST: 239.255.255.250  | TTL: 4 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 1901 | DPORT: 1900 | FLAGS:  - A - F - P |
ID: 57005 | SRC: 192.168.0.1 | DST: 239.255.255.250  | TTL: 4 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 1901 | DPORT: 1900 | FLAGS:  - A - F - P |
ID: 57005 | SRC: 192.168.0.1 | DST: 239.255.255.250  | TTL: 4 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 1901 | DPORT: 1900 | FLAGS:  - A - F - P |
ID: 43410 | SRC: 192.168.0.8 | DST: 192.168.61.104  | TTL: 128 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 58477 | DPORT: 53 | FLAGS:  - - - F - - |
ID: 43409 | SRC: 192.168.0.8 | DST: 192.168.61.104  | TTL: 128 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 49583 | DPORT: 53 | FLAGS:  - - - F - - |
ID: 10879 | SRC: 192.168.0.8 | DST: 8.8.8.8  | TTL: 128 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 49583 | DPORT: 53 | FLAGS:  - - - F - - |
ID: 10880 | SRC: 192.168.0.8 | DST: 8.8.8.8  | TTL: 128 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 58477 | DPORT: 53 | FLAGS:  - - - F - - |
ID: 1586 | SRC: 192.168.61.104 | DST: 192.168.0.8  | TTL: 253 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 53 | DPORT: 49583 | FLAGS:  - - - F - - |
ID: 1587 | SRC: 192.168.61.104 | DST: 192.168.0.8  | TTL: 253 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 53 | DPORT: 58477 | FLAGS:  - - - F - - |
ID: 43411 | SRC: 192.168.0.8 | DST: 192.168.61.104  | TTL: 128 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 64955 | DPORT: 53 | FLAGS:  - - - F - - |
ID: 1588 | SRC: 192.168.61.104 | DST: 192.168.0.8  | TTL: 253 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 53 | DPORT: 64955 | FLAGS:  - - - F - - |
ID: 43412 | SRC: 192.168.0.8 | DST: 192.168.61.104  | TTL: 128 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 64955 | DPORT: 53 | FLAGS:  - - - F - - |
ID: 47107 | SRC: 8.8.8.8 | DST: 192.168.0.8  | TTL: 59 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 53 | DPORT: 58477 | FLAGS:  - - - F - - |
ID: 10881 | SRC: 192.168.0.8 | DST: 8.8.8.8  | TTL: 128 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 64955 | DPORT: 53 | FLAGS:  - - - F - - |
ID: 1589 | SRC: 192.168.61.104 | DST: 192.168.0.8  | TTL: 253 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 53 | DPORT: 64955 | FLAGS:  - - - F - - |
ID: 25410 | SRC: 8.8.8.8 | DST: 192.168.0.8  | TTL: 113 | TOS: 0  | PROTOCOL: 17 | PROTO: UDP | SPORT: 53 | DPORT: 64955 | FLAGS:  - - - F - - |
```

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
Cache-Control: max-age=1900
Location: http://192.168.0.1:80/RootDevice.xml
NT: urn:schemas-upnp-org:device:WANConnectionDevice:1
USN:
uuid:upnp-WANConnectionDevice-1_0-28565adfad2f::urn:schemas-upnp-org:device:WANCo

NTS: ssdp:alive
Server: UPnP/1.0 UPnP/1.0 UPnP-Device-Host
```

```
4E 4F 54 49 46 59 20 2A 20 48 54 54 50 2F 31 2E 31 0D 0A 48 4F 53 54 3
A 20 32 33 39 2E 32 35 35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 0A
 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C 3A 20 6D 61 78 2D 61 67 65 3D
31 39 30 30 0D 0A 4C 6F 63 61 74 69 6F 6E 3A 20 68 74 74 70 3A 2F 2F 3
1 39 32 2E 31 36 38 2E 30 2E 31 3A 38 30 2F 52 6F 6F 74 44 65 76 69 63
 65 2E 78 6D 6C 0D 0A 4E 54 3A 20 75 72 6E 3A 73 63 68 65 6D 61 73 2D
75 70 6E 70 2D 6F 72 67 3A 64 65 76 69 63 65 3A 57 41 4E 43 6F 6E 6E 6
5 63 74 69 6F 6E 44 65 76 69 63 65 3A 31 0D 0A 55 53 4E 3A 20 75 75 69
 64 3A 75 70 6E 70 2D 57 41 4E 43 6F 6E 6E 65 63 74 69 6F 6E 44 65 76
69 63 65 2D 31 5F 30 2D 32 38 35 36 35 61 64 66 61 64 32 66 3A 3A 75 7
2 6E 3A 73 63 68 65 6D 61 73 2D 75 70 6E 70 2D 6F 72 67 3A 64 65 76 69
 63 65 3A 57 41 4E 43 6F 6E 6E 65 63 74 69 6F 6E 44 65 76 69 63 65 3A
31 0D 0A 4E 54 53 3A 20 73 73 64 70 3A 61 6C 69 76 65 0D 0A 53 65 72 7
6 65 72 3A 20 55 50 6E 50 2F 31 2E 30 20 55 50 6E 50 2F 31 2E 30 20 55
 50 6E 50 2D 44 65 76 69 63 65 2D 48 6F 73 74
```

Imagen 1. Pantalla de funcionamiento del packet sniffer

Como se muestra en la imagen 1, la pantalla superior muestra los paquetes escuchados, mostrando los siguientes parámetros:

- ID
- Fuente
- Destino
- TTS
- TOL
- Protocolo
- Puerto de salida
- Puerto destino
- Banderas propias del paquete

Los dos bloques inferiores están divididos en dos secciones, la sección izquierda muestra el contenido del paquete estructurado acorde al tipo de paquete que fue seleccionado. La pantalla inferior derecha desglosa el contenido en raw (crudo) del paquete seleccionado, es decir, previo a la interpretación de este, siguiendo una

estructura de contenido en hexadecimal con un tamaño variable acorde a lo enviado y el método de empaquetado que lleve el mismo.

Dentro de la imagen 1 también se aprecia los contenidos de la barra superior, en la que se puede cambiar el adaptador de red, de igual manera al realizar búsquedas por filtros o reiniciar el programa es necesario iniciar la captura de paquetes dentro del botón que indica captura. En el mismo botón de captura se puede seleccionar la detención, para pausar la escucha y registro de paquetes en la red,

Finalmente tenemos la sección de archivo, la cual generará un archivo con nombre y ubicación a elección del usuario en un formato de tabla, para poder ser abierto en Excel o cualquier software de archivos u hojas de cálculo compatibles.

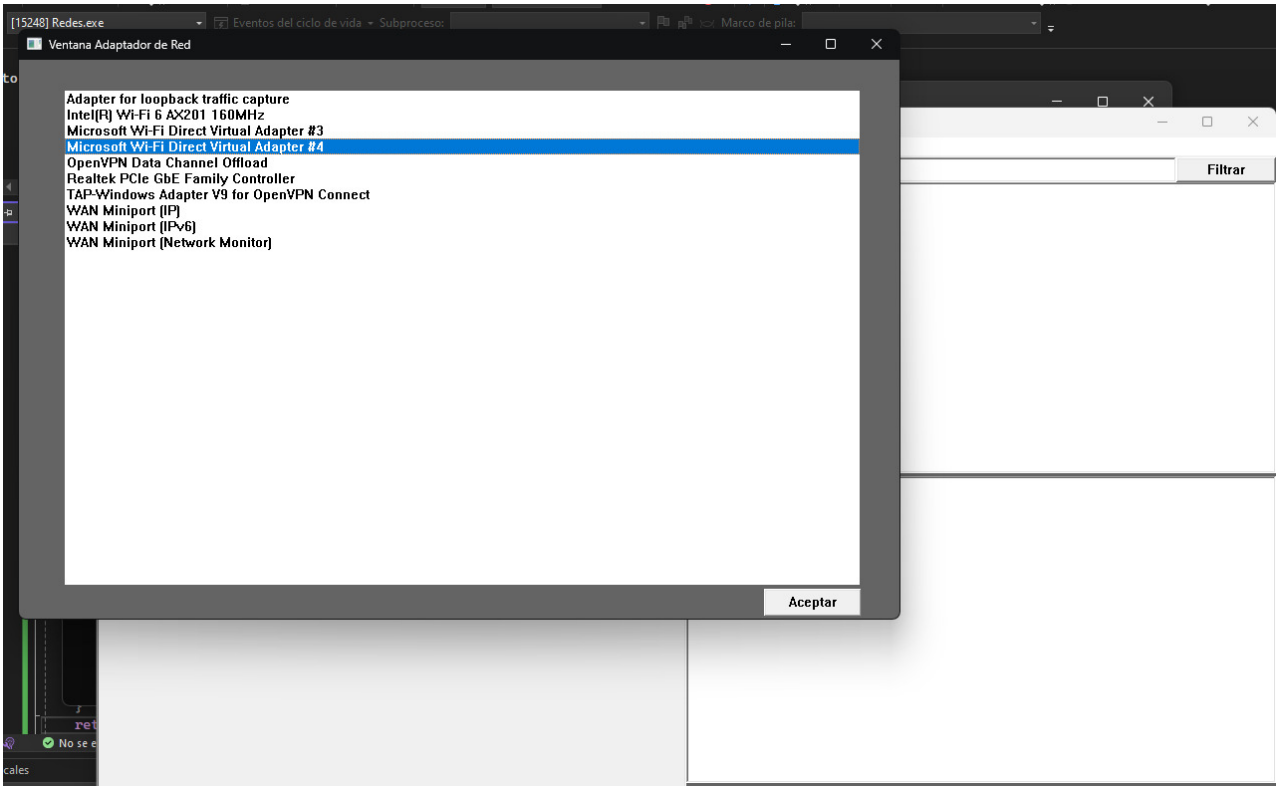


Imagen 2. Selección de adaptador de red

Para poder comenzar a escuchar una red, es necesario seleccionar el adaptador de red del computador que será utilizado, el packet sniffer detecta los adaptadores del dispositivo y genera una lista desde el cual el usuario seleccionará su elección para proceder con la escucha de la red desde el adaptador seleccionado.

Conclusiones:

Emilio Ortiz Romo: La importancia de la red en las últimas décadas ha aumentado en gran medida, por lo que es muy importante para muchas ramas de la tecnología el comprender y estudiar el trafico en las redes, por lo que herramientas como los packet sniffers de diversas fuentes resultan muy útiles. Como conclusión para este proyecto, se puede establecer que la misma importancia del análisis de diversas redes llevo a una mayor accesibilidad en cuanto al uso y creación de softwares de esta índole, mediante librerías como lo son npcap que permiten construir programas completos y funcionales siguiendo funciones sencillas de la misma librería. Finalmente considero importante mencionar que el uso de estas herramientas en un ámbito de seguridad es de gran importancia por lo que conocer la manera en la que estos trabajan puede favorecer el desarrollo personal y profesional en caso de seguir con el estudio de redes.

Uriel Rodríguez Guadarrama: El desarrollo y uso de un packet sniffer no solo permite profundizar en los aspectos técnicos del tráfico en redes, sino que también resalta la importancia de los principios éticos en la recopilación de datos. Comprender cómo estas herramientas funcionan ayuda a los profesionales a fortalecer la seguridad de sus sistemas, anticiparse a vulnerabilidades y promover un uso responsable de las tecnologías de monitoreo en redes.

Ricardo Almada Diaz: El análisis detallado de datos mediante un packet sniffer permite identificar patrones de tráfico, cuellos de botella y problemas de configuración en redes complejas. Esta capacidad facilita la implementación de estrategias de optimización que mejoran la eficiencia y velocidad de las comunicaciones, contribuyendo a entornos más robustos y adaptables para usuarios y empresas.

Ilse Jacqueline Martínez Espinosa: El diseño de un packet sniffer como proyecto académico o profesional fomenta el aprendizaje práctico de protocolos de comunicación y el funcionamiento interno de las redes. Esta experiencia refuerza habilidades clave en áreas como la programación, la resolución de problemas y la ciberseguridad, preparando a los desarrolladores para abordar retos tecnológicos actuales y futuros.

Código del Programa:

Referencias:

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

Stallings, W. (2017). Wireless communications and networks. *2022 International Conference On Smart Systems And Technologies (SST)*, 107. https://doi.org/10.1109/sst.2017.8188678

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/

NPCAP: Windows Packet Capture Library & Driver. (s. f.). https://npcap.com/

WinPcap: WinPcap tutorial: a step by step guide to using WinPcap. (s. f.). https://www.winpcap.org/docs/docs_411/html/group__wpcap__tut.html