Guía de Wireshark para Ingeniería en Redes y Telecomunicaciones

¿Qué es Wireshark? Wireshark es una herramienta de análisis de protocolo de red que permite capturar y examinar los datos que se transmiten a través de una red en tiempo real. Wireshark se utiliza para solucionar problemas de red, analizar y optimizar el rendimiento, así como para la seguridad informática.

¿Para qué sirve Wireshark?

Wireshark es útil para:

- **Administración de Redes**: Diagnosticar problemas de red, verificar configuraciones, y optimizar el rendimiento.
- **Ciberseguridad**: Detectar y analizar intentos de ataques, monitorear tráfico sospechoso y realizar análisis forenses de incidentes.

Elementos Claves para Comprender Wireshark

1. **Captura de Paquetes**: Wireshark captura todos los paquetes de datos que pasan por la red a la que está conectado.
2. **Interfaz de Usuario**: Permite ver los paquetes capturados y filtrarlos para centrarse en información específica.
3. **Decodificación de Protocolos**: Wireshark entiende y muestra los datos de diferentes protocolos de red.
4. **Filtros de Captura y Visualización**: Para enfocar el análisis en datos específicos.
5. **Análisis y Estadísticas**: Herramientas incorporadas para interpretar los datos y generar estadísticas útiles.

¿Cómo funciona Wireshark?

1. **Captura de Tráfico**: Conecta tu máquina a la red y usa Wireshark para comenzar a capturar el tráfico.
2. **Visualización de Paquetes**: Wireshark mostrará los paquetes capturados en una interfaz gráfica.
3. **Filtros**: Utiliza filtros para enfocar tu análisis en paquetes específicos.
4. **Análisis de Datos**: Examina los detalles de cada paquete, desde los encabezados hasta los datos contenidos.
5. **Guardado y Exportación**: Guarda las capturas para análisis posteriores o exporta los datos en diferentes formatos.

¿Qué se Debe Saber para Utilizar Wireshark?

- **Conocer Protocolos de Red**: Entender TCP/IP, UDP, HTTP, DNS, etc.
- **Uso de Filtros**: Cómo aplicar filtros de captura y visualización para centrarse en datos relevantes.
- **Interpretación de Datos**: Saber leer y entender los datos mostrados por Wireshark.
- **Resolución de Problemas**: Aplicar técnicas de diagnóstico para solucionar problemas de red.

Ejemplos de Uso de Wireshark

En Administración de Redes:

1. **Diagnóstico de Problemas de Conexión**:
    - Captura de tráfico para identificar pérdida de paquetes o latencias.
    - Uso de filtros para centrarse en el tráfico entre dos puntos específicos.
2. **Verificación de Configuración de Red**:
    - Monitorizar el tráfico DHCP para asegurarse de que los dispositivos están recibiendo configuraciones IP correctas.
    - Análisis de paquetes ARP para identificar conflictos de IP.
3. **Optimización de Rendimiento**:
    - Identificar cuellos de botella en la red.
    - Analizar el uso de ancho de banda por diferentes aplicaciones.

En Ciberseguridad:

1. **Detección de Escaneos de Nmap**:
    - Filtrar tráfico ICMP y TCP para identificar patrones de escaneo.
    - Análisis de puertos abiertos y servicios activos.
2. **Ataques de Fuerza Bruta**:
    - Monitorizar intentos de conexión fallidos.
    - Analizar tráfico SSH o RDP en busca de intentos repetidos de autenticación.
3. **Análisis de Claves Encriptadas por Telnet**:
    - Capturar tráfico Telnet.
    - Usar herramientas de descifrado para analizar datos en texto claro.

Filtros de Wireshark Más Utilizados

Administración de Redes

Propósito	Filtro Wireshark
IP Específica	`ip.addr == 192.168.1.1`
Protocolo Específico	`tcp`
Rango de Puertos	`tcp.port >= 20 && tcp.port <= 25`
Dirección MAC	`eth.addr == 00:0a:95:9d:68:16`
HTTP GET Requests	`http.request.method == "GET"`
DNS Requests	`dns`
Tráfico HTTPS	`ssl`
ARP Traffic	`arp`
Tráfico ICMP	`icmp`
DoS (SYN Flood):	`tcp.flags.syn == 1 and tcp.flags.ack == 0 and tcp.seq == 0`
Fuerza Bruta con Hydra (SSH)	`tcp.port == 22 and (ssh || tcp.flags.push == 1)`
Escaneo con Nmap (SYN Scan)	`tcp.flags.syn == 1 and tcp.flags.ack == 0`

Ciberseguridad

Propósito	Filtro Wireshark
Escaneo de Nmap	`tcp.flags.syn == 1 && tcp.flags.ack == 0`
Ataques de Fuerza Bruta	`tcp.flags == 0x02`
Tráfico SSH	`tcp.port == 22`
Análisis de Telnet	`tcp.port == 23`
Tráfico FTP	`ftp`
Búsqueda de Credenciales	`http contains "Authorization: Basic"`
Tráfico de Correo Electrónico	`smtp`
Beaconing de Malware	`ip.src == [IP_of_suspected_malware] && tcp.port == 443`
Paquetes Fragmentados	`ip.flags.mf == 1
Detección de Ataques DDoS	`tcp.flags.syn == 1 && tcp.flags.ack == 0 && frame.time_delta < 0.0001`

DoS (SYN Flood):	`tcp.flags.syn == 1 and tcp.flags.ack == 0 and tcp.seq == 0`
Fuerza Bruta con Hydra (SSH)	`tcp.port == 22 and (ssh || tcp.flags.push == 1)`
Escaneo con Nmap (SYN Scan)	`tcp.flags.syn == 1 and tcp.flags.ack == 0`