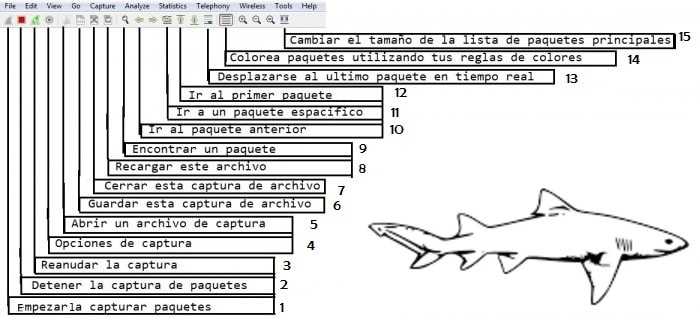
**Cómo funciona WireShark**

¿Qué hace?

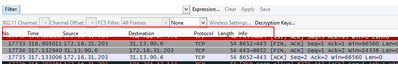
La herramienta intercepta el tráfico y lo convierte en un formato legible para las personas. Esto hace que sea más fácil identificar qué tráfico está cruzando la red, con qué frecuencia y la latencia que hay entre ciertos saltos. Si bien Wireshark admite más de 2.000 protocolos de red, muchos de ellos inusuales o antiguos, los profesionales encuentran una gran utilidad en el análisis de identidades IP. La mayoría de los paquetes son TCP, UPD e ICMP.

Dado el gran volumen de tráfico que atraviesa una red comercial típica, las utilidades de Wireshark ayudan a filtrarlo. Los filtros de captura solo recopilan los tipos de tráfico que le interesan al comercio y los de visualización le ayudan a acercarse al tráfico que quiere inspeccionar. El analizador de protocolo de red proporciona herramientas de búsqueda, que incluyen expresiones regulares y resaltado en color, para que sea más fácil encontrar lo que se está buscando.

**Controles WireShark**



**Campos de Wireshark**

****

**Nro:** este campo indica el número del paquete que se está capturando.

**Time:** indica el tiempo transcurrido en cada paquete capturado.

**Source:** acá se indica la dirección IP origen que está generando la conexión.

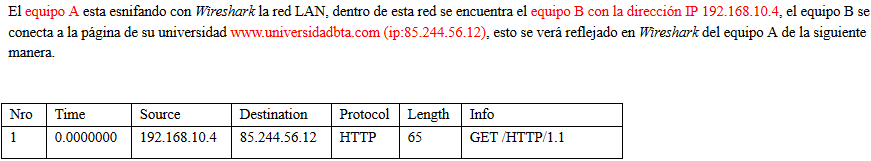
**Destination:** nos indica la dirección IP destino de la conexión.

**Protocol:** este campo nos indica el protocolo que se está utilizando para establecer una conexión.

**Length:** nos indica el tamaño del paquete.

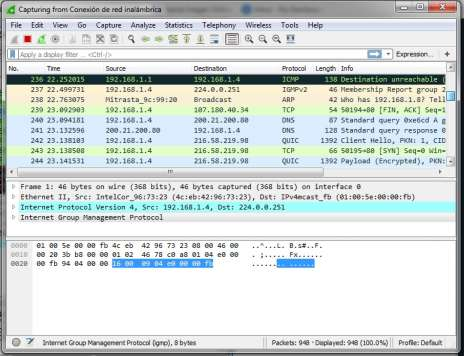
**Info:** este campo nos muestra información en concreto del comportamiento del protocolo utilizado.

**Ejemplo**

****

**Codificación de color en Wireshark**

Probablemente verá paquetes resaltados en verde, azul y negro. Wireshark utiliza colores para ayudarle a identificar los tipos de tráfico de un vistazo. De forma predeterminada, el verde es el tráfico TCP, el azul es el tráfico DNS, el azul claro es el tráfico UDP y el negro identifica los paquetes TCP con problemas; por ejemplo, podrían haber sido entregados dañados.

****

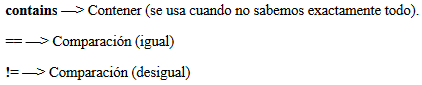
Para ver exactamente lo que significan los códigos de color, haga clic en View> Coloring rules. También puede personalizar y modificar las reglas de coloración de aquí, si lo desea(solo debe hacer doble clic sobre alguno de ellos y escoge el color que quiera).

**Comandos o filtros para Wireshark**

Concatenadores:



Operadores:

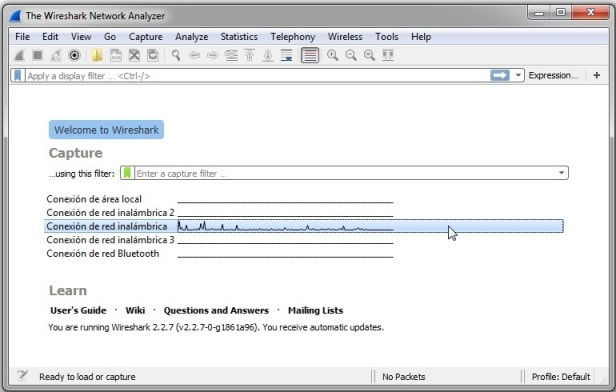


Protocolos:

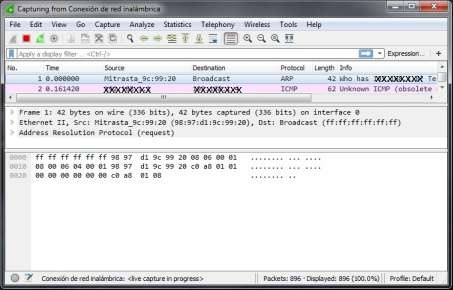
|  |  |
| --- | --- |
| Sintaxis | Significado |
| ssl | Protocolo SSL (capa segura). |
| telnet | Telnet |
| dns | DNS |
| msnms | Mensajería Instantánea (Messenger) |
| ftp | Protocolo FTP (se podría ver el nombre de usuario y contraseña) |
| dtp-data | Nos permite ver los datos del protocolo FTP |
| ip | Protocolo IP |
| ip.addr==93.184.216.34 | Permite examinar los paquetes de una IP especifica |
| ip.src==192.168.1.1 | Dirección IP origen |
| ip.dst==192.168.1.1 | Dirección IP destino |
| tcp | Protocolo TCP |
| tcp.port==80 | Indicamos los paquetes en el puerto deseado (en este caso 80) |
| tcp.srcport==80 | Indicamos el puerto de origen |
| tcp.dstport==80 | Indicamos el puerto de destino |
| http | Protocolo HTTP |
| http.host==”www.google.com” | Permite ver los paquetes que tengan a Google como host |
| http.date==”Wed, 30 Mar 2011 22:40:55 GMT” | Paquetes respecto a una fecha |
| http.content\_type==”aplication/json” | Muestra el contenido de los paquetes aplicación jason |
| http.content\_type==”image/jpeg” | Muestra el contenido de imágenes jpeg  Nota: otros archivos  \*text/html  \*text/css  \*video/quicktime  \*application/zip |
| http.request.method==”GET” | Permite examinar los metodos de peticion GET  \*POST |
| http.user\_agent contains “Mozilla” | Permite la obtención de peticiones realizadas desde un navegador determinado.  \*Gecko  \*Microsoft  \*Linux |
| http.server==”ngnix” | Visualiza los paquetes de respuesta del servidor hacia los clientes  \*sffe (Safer Server)  \*gws (Google Web Server)  \*Apache (Aquí mediante la palabra clave contains podemos diferenciar otros valores como PHP, Unix, etc)  \*PWS (Personal Web Server)  \*GFE (GFE/2.0 web server) |
| http.request.uri!=\* | No se visualizan los paquetes NOTIFY \* HTTP |

**Como usar wireshark para capturar paquetes**

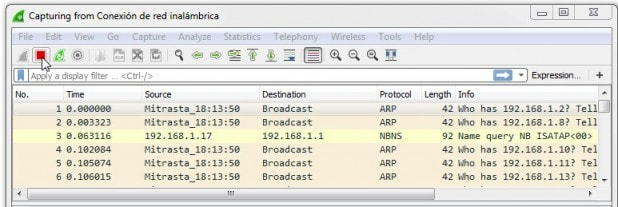
1. Hacer click sobre la interfaz para poder capturar paquetes de la misma.

****

1. Tan pronto como hagas clic en el nombre de la interfaz, verás que los paquetes empiezan a aparecer en tiempo real. Wireshark captura cada paquete enviado hacia o desde tu sistema. Si estás haciendo capturas en una interfaz inalámbrica y tienes el modo promiscuo activado en sus opciones de captura(el cual está habilitado por defecto), también verá los paquetes de los otros paquetes de la red.

****

1. Haz clic en el botón detener captura (Boton Rojo cuadrado #2) cerca de la esquina superior izquierda de la ventana cuando desees detener la captura de tráfico.

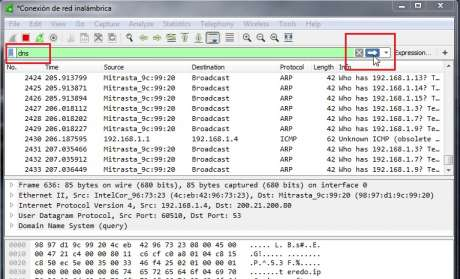


1. Luego de que haya presionado el botón Detener captura usted puede guardar esa captura para examinarla posteriormente, presionando Ctrl+Shift+S, o simplemente vaya ha File>Save as , y coloca un nombre de su preferencia, como por ejemplo PrimeraCaptura. Luego presiona Guardar.
2. Si no hay nada interesante en tu propia red para inspeccionar, Wireshark te tiene la solución. La página web de Wireshark contiene una página de archivos de captura de ejemplo que puede cargar e inspeccionar.
3. Abrir un archivo de captura es fácil; Simplemente, después de que tengas guardado cualquier paquete, cierre wireshark y vuélvalo a abrir, aparecerá de inmediato el paquete que capturó anteriormente, haz doble clic en el archivo y este se abrirá en la pantalla principal de Wireshark automáticamente, sino también puedes desde el menú de Wireshark File -> Open. También puedes guardar tus propias capturas en Wireshark y abrirlas más tarde.

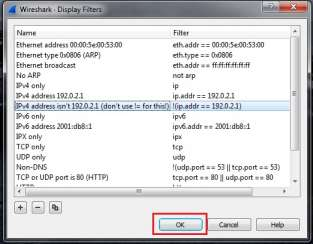
**Como usar WireShark para filtrar paquetes**

Si está intentando inspeccionar algo específico, como el tráfico que envía un programa al llamar a casa, ayude a cerrar todas las demás aplicaciones que utilizan la red para reducir el tráfico. Sin embargo, es probable que tenga una gran cantidad de paquetes para filtrar. Ahí es donde entran los filtros de Wireshark.

La forma más básica de aplicar un filtro es escribiéndola en el cuadro de filtro en la parte superior de la ventana y haciendo clic en Aplicar (o presionando Enter). Por ejemplo, escriba dns y verá sólo paquetes DNS. Cuando comience a escribir, Wireshark te ayudará a completar tu filtro automáticamente, luego también puedes darle clic en la flecha azul de la imagen de abajo para enfocarte en lo que buscas.

****

También puede hacer clic en el menú Analizar y seleccionar Mostrar filtros para crear un nuevo filtro.

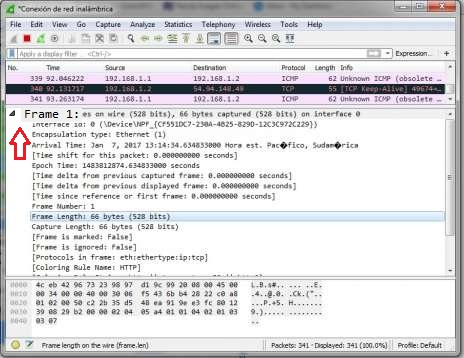


Otra cosa interesante que puedes hacer es hacer clic con el botón derecho del ratón en un paquete y seleccionar Follow -> TCP Stream (seguir la secuencia TCP).Verá la conversación completa entre el cliente y el servidor.

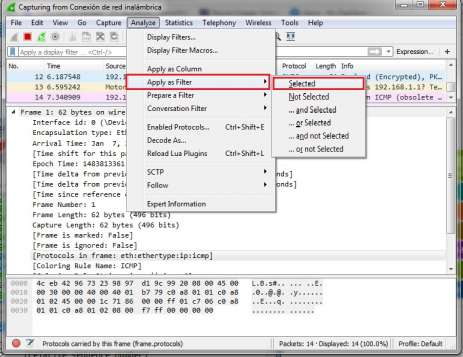


## **Cómo usar Wireshark para analizar paquetes**

1. Haz clic en un paquete para seleccionarlo y puedes bajar para ver sus detalles.



1. También puedes crear filtros desde aquí: haz clic derecho en uno de los detalles y usa el submenú Aplicar como filtro para crear un filtro basado en él.



**Cómo implementar DDoS**

Un ataque DDoS tiene como objetivo inhabilitar un servidor, un servicio o una infraestructura. Existen diversas formas de ataque DDoS: por saturación del ancho de banda del servidor para dejarlo inaccesible, o por agotamiento de los recursos del sistema de la máquina, impidiendo así que esta responda al tráfico legítimo.

Durante un ataque DDoS, se envían simultáneamente múltiples solicitudes desde distintos puntos de la red. La intensidad de este «fuego cruzado» compromete la estabilidad, y, en ocasiones, la disponibilidad del servicio.

DDos (que viene de Distributed Denial of Service) es un ataque (que se puede realizar de diversas formas) a un sistema de computadoras que causa que un servicio o recurso sea inaccesible a los usuarios legítimos.

Normalmente provoca la pérdida de la conectividad de la red por el **consumo del ancho de banda de la red** de la víctima o sobrecarga de los recursos del sistema de la víctima (incluso aunque tengas un VPS, por ejemplo).

Existen varios tipos de ataque DDos:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del ataque | Nivel OSI | Tipo de ataque | Explicación del ataque |
| *ICMP echo request flood* | L3 | Recursos | También denominado Ping Flood. Envío masivo de paquetes (ping), que implican una respuesta por parte de la víctima con el mismo contenido que el paquete de origen. |
| *IP Packet Fragment Attack* | L3 | Recursos | Envío de paquetes IP que remiten voluntariamente a otros paquetes que nunca se envían, saturando así la memoria de la víctima. |
| *SMURF* | L3 | Ancho de banda | Ataque por saturación ICMP que usurpa la dirección de origen para redirigir las múltiples respuestas hacia la víctima. |
| *IGMP Flood* | L3 | Recursos | Envío masivo de paquetes IGMP (protocolo de gestión de grupos de internet) |
| *Ping of Death* | L3 | Explotación | Envío de paquetes ICMP que explotan fallos del sistema operativo |
| *TCP SYN Flood* | L4 | Recursos | Envío masivo de solicitudes de conexión TCP |
| *TCP Spoofed SYN Flood* | L4 | Recursos | Envío masivo de solicitudes de conexión TCP usurpando la dirección de origen |
| *TCP SYN ACK Reflection Flood* | L4 | Ancho de banda | Envío masivo de solicitudes de conexión TCP a un gran número de máquinas, usurpando la dirección de origen por la dirección de la víctima. En ancho de banda de la víctima queda saturada por las respuestas a dichas peticiones. |
| *TCP ACK Flood* | L4 | Recursos | Envío masivo de acuses de recibo de segmentos TCP |
| *TCP Fragmented Attack* | L4 | Recursos | Envío de segmentos TCP que remiten voluntariamente a otros que nunca se envían, saturando la memoria de la víctima |
| *UDP Flood* | L4 | Ancho de banda | Envío masivo de paquetes UDP (sin necesidad de establecer conexión previa) |
| *UDP Fragment Flood* | L4 | Recursos | Envío de datagramas que remiten voluntariamente a otros datagramas que nunca se envían, saturando así la memoria de la víctima |
| *Distributed DNS Amplification Attack* | L7 | Ancho de banda | Envío masivo de peticiones DNS usurpando al dirección de origen de la víctima hacia un gran número de servidores DNS legítimos. Como la respuesta tiene un mayor volumen que la pregunta, el ataque se amplifica |
| *DNS Flood* | L7 | Recursos | Ataque de un servidor DNS mediante el envío masivo de peticiones |
| *HTTP(S) GET/POST Flood* | L7 | Recursos | Ataque de un servidor web mediante el envío masivo de peticiones |
| *DDoS DNS* | L7 | Recursos | Ataque de un servidor DNS mediante el envío masivo de peticiones desde un gran número de máquina s controladas por el atacante |

Es posible realizar un ataque con ayuda de la la herramienta Low Orbit Ion Cannon. que es es una aplicación diseñada para realizar un ataque de denegación de servicio desarrollada por «praetox» usando el lenguaje de programación C# (Existe también un fork en C++ y Qt llamado LOIQ). La aplicación realiza un ataque de denegación de servicio del objetivo enviando una gran cantidad de paquetes TCP, paquetes UDP o peticiones HTTP con objeto de determinar cuál es la cantidad de peticiones por segundo que puede resolver la red objetivo antes de dejar de funcionar.

Se recomienda esconder la IP antes de realizar un ataque, para evitar ser rastreados. El primer paso es introducir la **IP o la URL del servidor** donde se realizará el ataque y seleccionarla como objetivo. Podemos seleccionar el **puerto, tiempo, protocolo (TCP, UDP, HTTP) y si queremos esperar respuesta del servidor.** El siguiente paso es escribir el **mensaje TCP/UDP que queremos enviar con los paquetes y la velocidad a la que serán enviados**. Una vez seleccionados los parámetros, simplemente seleccionamos enviar carga. Se comenzarán a enviar miles de peticiones a la IP seleccionada, con ayuda de Wireshark podemos visualizar las peticiones, su origen y el protocolo que usan.

