wireshark, monitorización, sniffer

Wireshark

Wireshark antes conocido como ethereal es un programa que permite analizar nuestra red es busca de problemas

Instalación

Para instalarlo en ubuntu basta con

```
sudo apt-get install wireshark
```

Una vez instalado tenemos que ejecutarlo como root para ello desde la línea de comando

```
sudo wireshark
```

o

```
su wireshark
```

o

```
gksu -u root /usr/bin/wireshark
```

Nos saldrá un mensaje advirtiendonos de que es peligroso ejecutar wireshark como root le damos ok y listo

aunque sale un mensaje de advertencia es necesario ejecutar el programa como root ya que si no, no sale el listado de interfaces ni tampoco se puede capturar (http://wiki.wireshark.org/CaptureSetup/CapturePrivileges)

Preferencias

Todos los ajustes que pueden realizarse en la interfaz de Wireshark están en el menú Edit→ Preferences. **(CTRL + Mayúsculas + P)**

Es posible concentrar un conjunto de ajustes a la interfaz de Wireshark en un perfil, con el objetivo de aplicarlos con un solo click, solo cuando sea necesario.

Wireshark mantiene un conjunto de carpetas simples para almacenar los ajustes personales, y aquellos que vienen por defecto con la aplicación.

Ajustes de TCP (Prefeences->Protocols->TCP)

- **Allow subdissector to reassemble TCP streams:** Seleccionar para lograr descargar los archivos (fotos, documentos, etc) capturados. Deseleccionar, para ver los comandos del protocolo de aplicación que son parte de la cabecera.
- **Track number of bytes in flight:** Seleccionar, se lleva el conteo de los paquetes TCP que aún no han sido confirmados (ACK).
- **Calculate conversation timestamps:** Seleccionar, se muestra el tiempo entre paquetes por cada conversación TCP.

Filtros

Ejemplos

- tcp.srcport == 34 ←Capturamos sólo los paquetes con puerto origen 34
- icmp[0:1] == 08 ← paquetes del tipo echo request
- frames contains “@miempresa.es” ← paquetes de correo
- frame contains “password” || frame contains “user”← para buscar contraseñas
- (tcp.port eq 445) and !(ip.addr eq 192.168.1.1) ←muestra el tráfico tcp 445 minetras filtra una dirección. Esto es útil para detectar por ejemplo el virus Conficker. Buscamos todo ese tráfico pero filtramos por ejemplo nuestro servidor de dominio que es normal que si use el puerto 445
- frame.time_delta >1 = paquetes que tardaron más de 1 sg

Eventos SMB

- Close Request (0x04) Cerrado de archivo.
- NT Create Andx Request (0xa2) Creación de carpetas y arhivos.
- smb.cmd simplmente filtrndo por smb.cmd sin valor alguno podemos analizar las transacciones SMB, errores, etc.
- SMB Rename Request (0x07) Renombrado de archivos.
- SMB Read AndX Request (0x2e) Apertura de un archivo.
- SMB Write Andx Request (0x2f) Escrituraen un archivo.
- SMB Cancer Request (0xa4)
- SMB Delete Request (0x06)

Ejemplo

- tshark -R “smb.cmd==0x2f”

Perfiles

Un perfil nos permite aplicar cambios sobre la interfaz de wireshark de forma inmediata. Deberíamos de crear varios perfiles según el tipo de tráfico que vamos a analizar. Para crear/modificar/eliminar perfiles →Menú Edit → Configuration Profiles o con **CTRL + Mayúsculas + A**

IO Graph

Filtros:

- tcp.analysis.lost_segment Perdida de paquetes o segmentos
- tcp.analysis.retransmission Mecanismo de rentransmision
- tcp.analysis.fast.retransmission Mecanismo de rentransmision rápida
- tcp.analysis.duplicate_ack Análisis de ACKs duplicados

Detección de problemas de Red

Para buscar errores en el menú Analyze → Expert Info composite . Desde ahí podemos acceder rápidamente a los distintos problemas.

Ojo El Wireshark marca como error unos pequeños paquetes que se repiten varias veces en el tiempo y que son usados por algunas aplicaciones que usan paquetes TCP Keep-Alive para asegurarse de que no se corta la conexión entre cliente y servidor. Por ejemplo con aplicaciones de consultas en Bases de Datos. Normalmente no es señal de problemas de red.

También es útil añadir las siguientes columnas al wireshark:

- Delta Time: Esta columna es necesaria para medir los tiempos de respuesta, retardos, etc
- Cumulative Bytes: Muestra la cantidad de datos enviados. si lo dividimos por el tiempo tardado en enviarlo obtenemos el rendimiento.
- TCP Windows Size: Útil si no sabemos el tamaño de la ventana TCP
- IP DSCP Value: Útil cuando monitorizamos tráfico VoIP. Permite ver entre otras cosas si Qos está configurado

Algunos errores típicos

- TCP Previous segmento lost nos indica que un segmento TCP anterior ha fallado
- Un TCP Dup ACK significa que el hay perdida de paquetes. si el número de paquetes perdidos es alto indica una latencia alta

Por regla general:

- Solo se esperan hasta 3 ACKs duplicados.
- Uno o dos ACKs duplicados indica una reordenación de los segmentos.
- Tres o más ACKs duplicados indica que se perdió el paquete.

Latencia

Según la wikipedia “la latencia de red es la suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red”. Es decir , la latencia es el tiempo que tarda una petición en completarse. Por ejemplo si estamos visualizando una página web, sería el tiempo desde que enviamos la solicitud hasta que la tenemos

cargada.

La latencia puede estar originada en La red, el Origen o el Destino

- **Latencia en la Red** → Se da cuando el tiempo de repuesta del destino (SYNC/ACK) a una petición SYNC del cliente, es alto.
- **Latencia en Origen (cliente)**→ Es cuando el tiempo es alto entre que el cliente envía un **ACK** y el cliente envía un **(Request)**
- **Latencia en Destino(servidor)→Se da cuando el tiempo entre que el destino envía un** ACK** y el destino envía la respuesta es elevado

Para ver problemas de latencia tenemos que habilitar la columna Delta Time

Referencias

- http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/cert_inf_seguridad_analisis_trafico_wireshark.pdf
- http://seguridadyredes.nireblog.com/post/2008/02/14/analisis-de-red-con-wireshark-interpretando-los-datos
- http://thenetworkzone.blogspot.com/2009/10/resolve-network-problems-faster-with.html
- https://www.academiaredes.com