***Trabajo Práctico*** *6*

*Análisis de tráfico TCP y UDP en GNU/Linux*

*Autores*

*Esteban Nicolas Andrada:*

[*nico86gnr@gmail.com*](mailto:nico86gnr@gmail.com)

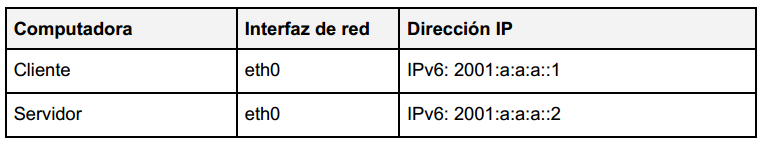
*Federico Sepulveda:* [*federico.sepulveda@alumnos.unc.edu.ar*](mailto:federico.sepulveda@alumnos.unc.edu.ar)

# Desarrollo

# Diagrama



**Tabla de asignación de direcciones IPv6**

****

**Servidor DNS con administración remota a través de WEBMIN**

**1.1.- Utilizando docker compose, crear e iniciar un container para ser utilizado como**

**servidor DNS**

**- Utilizaremos la siguiente imagen para crear el containter. La misma viene con “bind”**

**como servicio DNS y “webmin” para administración remota.**

**- image: "sameersbn/bind:latest"**

**- Se debe setear la variable de entorno “ROOT\_PASSWORD=ubuntu” en el container**

**para tener acceso a webmin**

**- Configurar WEBMIN para poder ser accedido a través de IPv6**



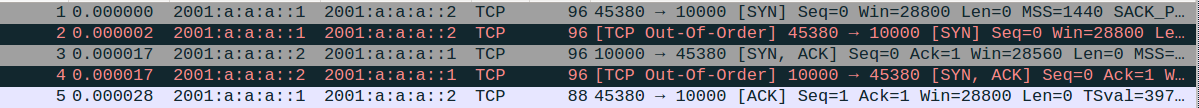
# Análisis de tráfico

# 2.1.- Comenzar a analizar el tráfico HTTPS con Wireshark en la interfaz que nos conecta con el servidor DNS.

# 2.2.- Desde el cliente utilizar el navegador web para poder autenticarse en el portal de WEBMIN. Ingresar utilizando la IPv6 del servidor. Luego cierre sesión y por último cierre la pestaña del navegador.

# 2.3- Analizar el tráfico en capa de enlace y responder:

## - ¿Como es la secuencia de iniciación de sesión de tcp? Indicar con capturas de pantalla donde se observen los mensajes



En la primera etapa, la máquina originadora (el cliente) transmite un segmento al puerto 10000 donde el indicador SYN está fijado en 1 (para indicar que es un segmento de sincronización), con número de secuencia N llamado número de secuencia inicial del cliente. En la segunda etapa, la máquina receptora (el servidor) recibe el segmento inicial que viene del cliente y luego le envía un acuse de recibo, que es un segmento en el que el indicador ACK está fijado en 1 y el indicador SYN está fijado en 1 (porque es nuevamente una sincronización). Este segmento incluye el número de secuencia de esta máquina (el servidor), que es el número de secuencia inicial para el cliente. El campo de mayor importancia en este segmento es el de acuse de recibo que contiene el número de secuencia inicial del cliente incrementado en 1. Por último, el cliente transmite un acuse de recibo, que es un segmento en el que el indicador ACK está fijado en 1 y el indicador SYN está fijado en 0 (ya no es un segmento de sincronización). Su número de secuencia ha sido incrementado y el acuse de recibo indica el número de secuencia inicial del servidor incrementado en 1.

Después de esta secuencia con tres intercambios, las dos máquinas están sincronizadas y la comunicación puede comenzar.

## - ¿Como es la secuencia de finalización de sesión de tcp? Indicar con capturas de pantalla donde se observen los mensajes.

La forma estándar de cerrar sesiones TCP es enviar un paquete FIN, luego esperar una respuesta FIN de la otra parte. Sin embargo ¿qué sucede cuando una parte de la conversación se rompe? Es por eso que existe el paquete Restablecer (RST).

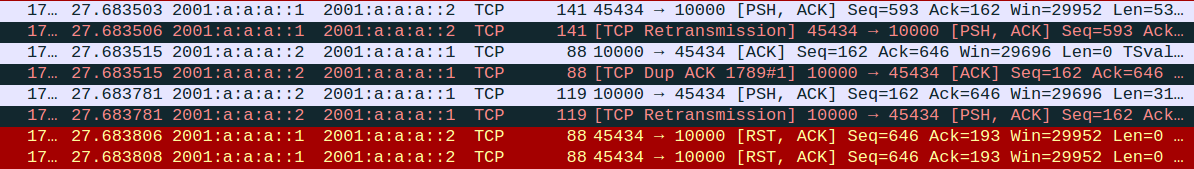
Las terminaciones anormales (como una desconexión) podrían aparecer debido a:

* La falta de recursos o interrupción de la red
* Un bloqueo / error durante la sesión
* Mientras que una parte ha cerrado la conexión la otra parte continúa enviando datos
* el servidor se niega a abrir una nueva conexión con el cliente

A veces, los paquetes RST son bastante "normales". Por ejemplo, cuando el usuario interrumpe manualmente una enorme transferencia de datos. La sesión TCP está enviando paquetes lo más rápido posible, de modo que cuando el cliente envía el FIN y cierra su parte, el servidor aún está enviando muchos datos por un momento. En este caso, el cliente envía paquetes RST hasta que el servidor deja de enviar datos.

También es importante tener en cuenta que algunas aplicaciones no cierran sesiones correctamente y simplemente usan un RST para cerrar cada sesión. Si bien esta no es una buena práctica, debe tener en cuenta que algunas aplicaciones se desarrollan de esta manera.

En conclusión una conexión TCP puede ser más complejo que abrir uno. La sesión puede cerrarse con un FIN doble, con una combinación de FIN + RST o solo con paquetes RST. Sin embargo, los paquetes RST también se pueden enviar sin ninguna conexión.



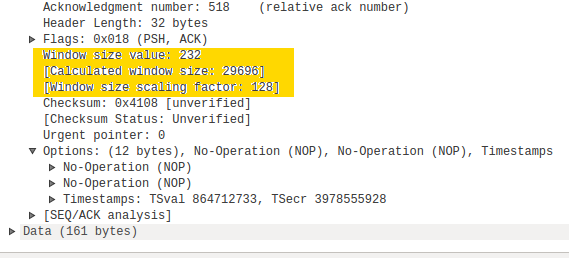
## - ¿Que utilidad tiene el tamaño de ventana en la sesión tcp? ¿Como se calcula? Mostrar un ejemplo mediante captura de pantalla de un segmento tcp

El tamaño de la ventana de recepción TCP es la cantidad de datos recibidos (en bytes) que pueden ser metidos en el buffer de recepción durante la conexión. La entidad emisora puede enviar una cantidad determinada de datos pero antes debe esperar un asentimiento con la actualización del tamaño de ventana por parte del receptor.

Un ejemplo sería el siguiente: un receptor comienza con un tamaño de ventana x y recibe y bytes, entonces su tamaño de ventana será (x - y) y el transmisor sólo podrá mandar paquetes con un tamaño máximo de datos de (x - y) bytes. Los siguientes paquetes recibidos seguirán restando tamaño a la ventana de recepción. Esta situación seguirá así hasta que la aplicación receptora recoja los datos del buffer de recepción. Para una mayor eficiencia en redes de gran ancho de banda, debe ser usado un tamaño de ventana mayor. El campo TCP de tamaño de ventana controla el movimiento de datos y está limitado a 16 bits, es decir, a un tamaño de ventana de 65.535 bytes.

Como el campo de ventana no puede expandirse se usa un factor de escalado. La escala de ventana TCP (*TCP window scale*) es una opción usada para incrementar el máximo tamaño de ventana desde 65.535 bytes, a 1 Gigabyte.

La opción de escala de ventana TCP es usada solo durante la negociación en tres pasos que constituye el comienzo de la conexión. El valor de la escala representa el número de bits desplazados a la izquierda de los 16 bits que forman el campo del tamaño de ventana. El valor de la escala puede ir desde 0 (sin desplazamiento) hasta 14. Hay que recordar que un número binario desplazado un bit a la izquierda es como multiplicarlo en base decimal por 2.



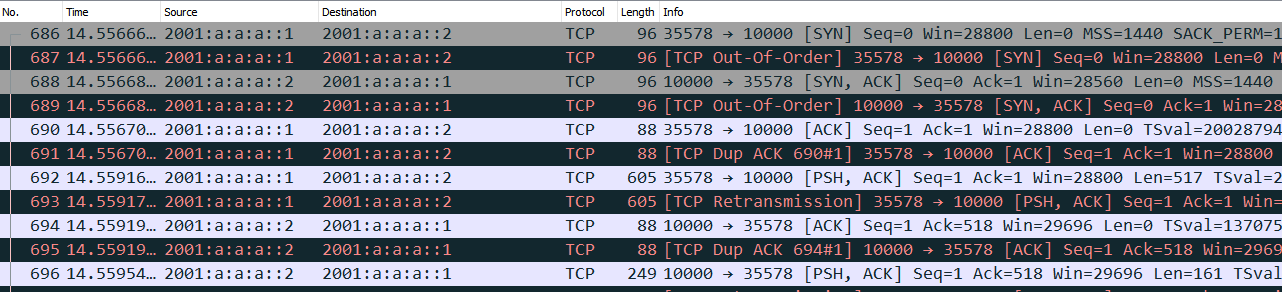
Calculated windows size = window size value \* window size scaling factor

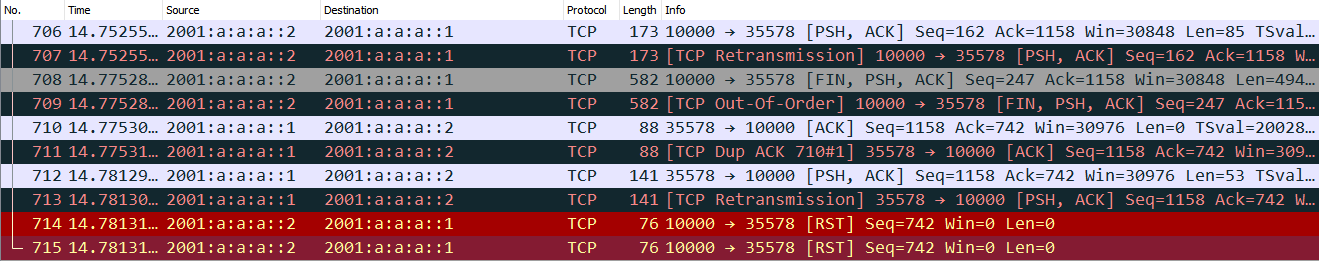
## - ¿Que significa que el tamaño de ventana sea cero?

Si el búfer de recepción está lleno, el sistema receptor anuncia un tamaño de ventana igual a cero y el sistema remitente no podrá seguir enviando datos. Una vez que la aplicación receptora recupera los datos del búfer de recepción, el sistema receptor podrá responder con un tamaño de ventana igual a la cantidad de datos leídos. A continuación, el TCP de sistema remitente podrá reanudar el envío de datos.

## - Como puede observarse en wireshark, durante el tiempo que tuvo abierto WEBMIN en su navegador, se iniciaron y cerraron múltiples sesiones tcp. Filtre alguna de estas sesiones y observe el comportamiento del tamaño de ventana durante la sesión.

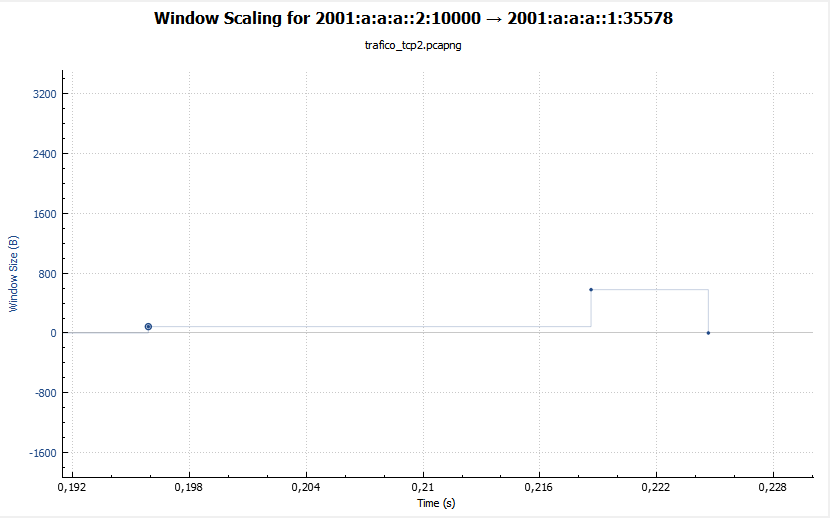
Se eligió la sesión que va desde 686 a 715 en número de paquetes. El sentido de la comunicación es desde el servidor al cliente





Se hizo capturas de los tamaños de ventanas como varían estos en el tiempo desde que se inicia la sesión hasta que esta finaliza

#### 



Se puede observar como la venta cambia de tamaño en el tiempo

#### BONUS

**3.1.- Configurar WEBMIN para que trabaje sin ssl**

**3.2.- Analizar tráfico mientras se ingresa a WEBMIN y se autentica**

**3.3.- En los paquetes capturados identificar el nombre de usuario y contraseña de WEBMIN**

### Ejercicio 2: Análisis de tráfico UDP sobre servidor DNS

**1.1.- Configurar el contenedor para que almacene los archivos contenidos en /data de manera persistente.**

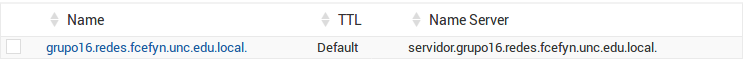
Los volúmenes de datos son directorios del host montados en un directorio del contenedor, aunque también se pueden montar ficheros. Cuando borramos el contenedor, no se elimina el volumen asociado. Enlazamos un directorio local con un directorio dentro del contenedor especificando el modo de acceso

****

**1.2.- Desde WEBMIN configurar como servidor maestro de DNS para el dominio:**

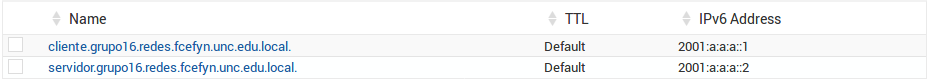
* **grupoX.redes.fcefyn.unc.edu.local**

**Donde X es el número de grupo.**

****

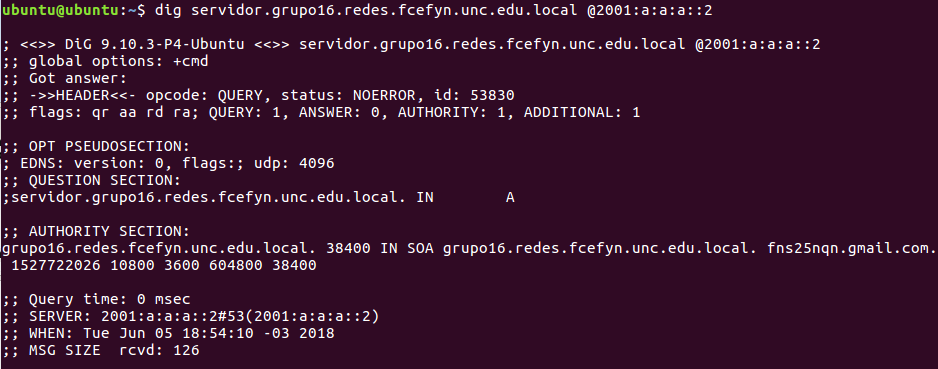
**1.3.- Agregar los registros AAAA para las IPs de servidor y cliente:**

* **servidor.grupoX.redes.fcefyn.unc.edu.local**
* **cliente.grupoX.redes.fcefyn.unc.edu.local**

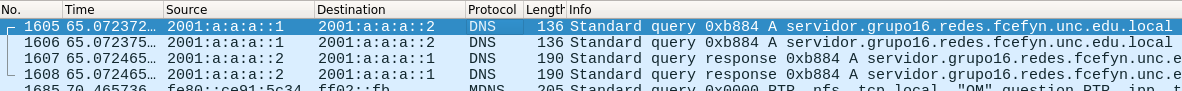
****

##### Análisis de tráfico

**2.1.- Realizar consultas DNS desde una máquina cliente utilizando el comando “dig”**

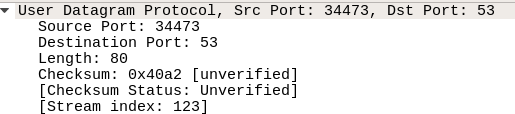
****

**2.2.- Analizar el tráfico DNS con Wireshark. Identificar cada campo de las Capas 4 y 5. Responder:**

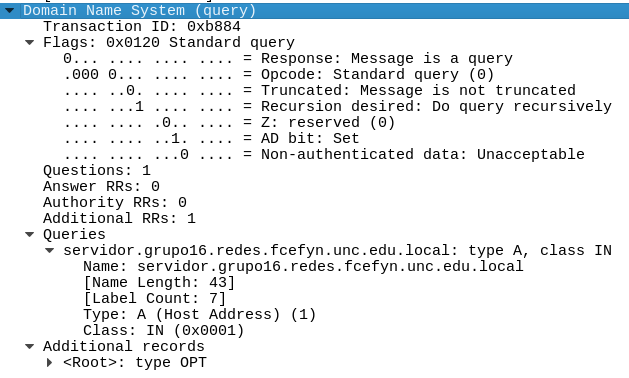
****

* **¿Que campos posee la cabecera UDP? indicar en print de pantalla de wireshark**

Capa 4:

****

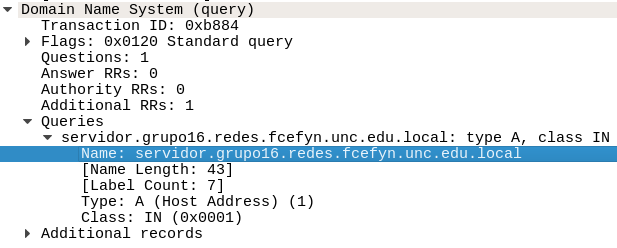
Capa 5:



* **Analizar la información transportada en el segmento UDP:**
  + **¿A qué capa del stack tcp/ip pertenece dicha información?**

La información transportada por UDP es el protocolo DNS el cual es capa 5 (aplicación)

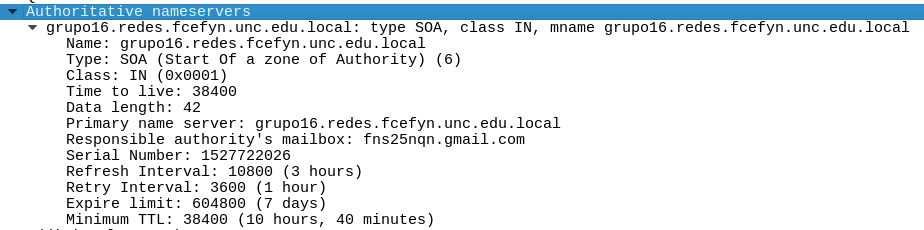
* + **Indicar en un print de pantalla el nombre de dominio del cual se quiere conocer su IPv6**

****

* + **¿Que es una zona de autoridad?**

Permite al servidor maestro cargar la información de una zona. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad. La información de zona de autoridad es almacenada de forma local en un fichero en el servidor DNS. Este fichero puede incluir varios tipos de registros, por ejemplo el registro AAAA el cual es un registro de dirección que resuelve el nombre de un anfitrión hacia una dirección IPv6 de 128 bits.

* + **¿Cual es la zona de autoridad de nuestro servidor dns? indicarlo mediante print de pantalla de wireshark, en el mensaje de respuesta**

****

* + **¿Qué sucede si el paquete nunca llega a destino? ¿el transmisor envía dicho paquete?**

En este caso el paquete no se retransmite, porque el emisor no se entera que el paquete no llegó a destino ya que se trata del protocolo UDP y este no es orientado a conexión. Se pudo observar en este práctico que por cada consulta se envían dos paquetes con el mismo contenido y en respuesta se reciben dos paque también con el mismo contenido.