

TP FINAL INTEGRADOR

Autor : Axel Tomás Resnik

Legajo : 190168

Startup usados :

Datos Startup:

```
chown www-data:www-data /usr/lib/cgi-bin/datos.pl
chmod 750 /usr/lib/cgi-bin/datos.pl
sed -i 's/\KeepAlive On/KeepAlive Off/' /etc/apache2/apache2.conf
a2enmod cgid
```

```
# IP de la interfaz
ip a a 12.12.0.12/24 dev eth0

# Levanto las Interfaces
ip link set dev eth0 up

# tabla de ruteo
ip route a default via 12.12.0.1

# inicio apache
service apache2 start
```

DNS startup

```
chmod 755 /etc
chmod 755 /etc/bind

# ip interfaces
ip a a 8.8.8.8/28 dev eth0

# levanto las interfaces
ip link set dev eth0 up

# rutas
ip route a default via 8.8.8.1

# inicio servicio bind
service bind start
```

PCUSUARIO startup

```
source /root/.bashrc
# IP interfaces
ip addr add 10.10.0.10/24 dev eth0

# levantar Interfaces
ip link set eth0 up

# rutas
```

```
ip route add 8.8.8.8 via 10.10.0.100
```

```
# Configurar DNS
```

```
echo "nameserver 8.8.8.8" > /etc/resolv.conf
```

```
# él export http_proxy="http://10.10.0.1:3128" lo hice en la consola una vez ejecutado él entorno
```

PROXY startup:

```
echo "visible_hostname proxy.tpfinal-tyr.com" >> /etc/squid/squid.conf
```

```
sed -i 's/\#http_access allow localnet/http_access allow all/'
```

```
/etc/squid/squid.conf
```

```
#rm -r /var/spool/squid/*
```

```
#squid -z
```

```
# IP interfaces
```

```
ip a a 10.10.0.1/24 dev eth0
```

```
# levanto las interfaces
```

```
ip link set dev eth0 up
```

```
# rutas
```

```
ip route a default via 10.10.0.100
```

```
# configuro el resolver
```

```
echo "nameserver 8.8.8.8" > /etc/resolv.conf
```

```
# inicio squid
```

```
service squid start
```

ROUTER1 startup:

```
# IP interfaces
```

```
ip a a 190.7.231.40/24 dev eth0
```

```
ip a a 10.10.0.100/24 dev eth1
```

```
# levanto las interfaces
```

```
ip link set dev eth0 up
```

```
ip link set dev eth1 up
```

```
# rutas
```

```
ip route a 8.8.8.0/28 via 190.7.231.30
```

```
ip route a 12.12.0.0/24 via 190.7.231.20
```

ROUTER2 startup:

```
# IP interfaces
```

```
ip a a 190.7.231.30/24 dev eth0
```

```
ip a a 8.8.8.1/28 dev eth1
```

```
# levanto las interfaces
ip link set dev eth0 up
ip link set dev eth1 up

# rutas
ip route a 10.10.0.0/24 via 190.7.231.40
ip route a 12.12.0.0/24 via 190.7.231.20
```

ROUTER3 startup:

```
# IP interfaces
ip a a 190.7.231.20/24 dev eth0
ip a a 12.12.0.1/24 dev eth1

# levanto las interfaces
ip link set dev eth0 up
ip link set dev eth1 up

# rutas
ip route a 8.8.8.0/28 via 190.7.231.30
ip route a 10.10.0.0/24 via 190.7.231.40
```

WWW startup:

```
sed -i 's/\KeepAlive On/KeepAlive Off/' /etc/apache2/apache2.conf

# IP interfaces
ip a a 12.12.0.11/24 dev eth0

# levanto las interfaces
ip link set dev eth0 up

# rutas
ip route a default via 12.12.0.1

# ejecuto el servicio apache
service apache2 start
```

Ejercicio 1.

En el excel está todas las pdu ordenadas

Ejercicio 2.

PROTOCOLO	CAPA	CANTIDAD DE PDU	EXPLICACION
HTTP	APLICACIÓN (5)	12	Se encarga de intercambiar información entre el cliente y el servidor, es capa aplicación por que no se preocupa como llegan los datos al otro extremo e interactua directamente con el usuario
DNS	APLICACIÓN (5)	18	Traduce nombre a direcciones ip, es un protocolo que interactua directamente con el usuario
UDP	TRANSPORTE(4)	18	
TCP	TRANSPORTE(4)	125	Proporciona una conexión confiable de extremo a extremo controlando errores, flujo, orden y congestión, interactua directamente con la interfaz del destinatario sin importarle como llegaron los segmentos
IPv4	RED(3)	161	Se encarga de conmutar los paquetes por la red para que lleguen al destino, trata de enviar los paquetes sin controlar errores ni el orden. Es capa red por que contiene toda la logica para poder enrutar los paquetes por la redes
ICMP	RED(3)	6	Se encarga de realizar control en la capa de red como enviar errores (ttl expirado) o hacer ping entre 2 host
ARP	ENLACE(2)	14	Traduce direcciones ip a direcciones MAC a nivel local (red local), Es capa enlace por que funciona a nivel local, osea, a nivel enlace
ETHERNET	ENLACE(2)	169	Define cómo se empaquetan y envían los datos entre nodos directamente conectados. Trabaja a nivel local y controla el acceso al medio

Ejercicio 3.

	CONEXIÓN TCP HOST A PROXY	CONEXIÓN TCP PROXY A WWW	CONEXIÓN TCP HOST A PROXY 2	CONEXIÓN TCP PROXY A DATOS	CONEXIÓN TCP HOST A PROXY 2	CONEXIÓN TCP PROXY A DATOS
socket Cliente	(10.10.0.10, 58146)	(10.10.0.1, 60866)	(10.10.0.10, 58156)	(10.10.0.1, 35260)	(10.10.0.10, 58158)	(10.10.0.1, 35264)
socket servidor	(10.10.0.1, 3128)	(12.12.0.11, 80)	(10.10.0.1, 3128)	(12.12.0.12, 80)	(10.10.0.1, 3128)	(12.12.0.12, 80)
segmentos apertura	3,4,5	30,31,36	73,74,75	86,87,92	128,129,130	133,135,139
segmento cierre	70,71,72	52,60,61	120,121,127	98,110,111	161,162,163	154,164,168
finalidad	realizar una solicitud de una pagina al proxy (www.tpfinal-tyr.com)	solicitud get de la pagina principal ubicada en la raiz del sitio (/)	realizar una solicitud de una pagina al proxy (datos.tpfinal-tyr.com)	solicitud get del archivo index.html	realizar una solicitud get de un archivo datos.pl ubicado en el directorio cgi-bin del servidor datos	Solicitud get del archivo datos.pl pedido por el host
Parametros intercambiados en apertura	MSS: 1460 Sack Permitted Window scale: 7	MSS: 1460 SACK permitted Window scale: 7	MSS: 1460 SACK permitted Window scale: 7	MSS: 1460 SACK permitted Window scale: 7	MSS: 1460 SACK permitted Window scale: 7	MSS: 1460 SACK permitted Window scale: 7

Ejercicio 4.

NUMERO PDU	PROTOCOLO	CAPA	FINALIDAD
73	TCP	TRANSPORTE(4)	El host Inicia una solicitud de conexion con el proxy(Bandera SYN activada)
74	TCP	TRANSPORTE(4)	El proxy confirma la solicitud con la bandera de ack activada y envia su propia solicitud de conexión al host activando tambien la bandera SYN
75	TCP	TRANSPORTE(4)	El host confirma la solicitud de apertura de conexión del proxy enviando un ACK
76	HTTP	APLICACION(5)	El host realiza una solicitud GET a datos.tpfinal-tyr.com pidiendo el archivo index.html
77	TCP	TRANSPORTE(4)	El proxy confirma la recepcion del GET enviando un ACK
116	TCP	TRANSPORTE(4)	El proxy envia los encabezados al host con la bandera PSH y ACK activadas
117	TCP	TRANSPORTE(4)	El host confirma la recepcion de los datos enviando un ACK
118	HTTP	APLICACION(5)	El proxy envia el 200 OK que le respondio el servidor datos junto al contenido del archivo index.html
119	TCP	TRANSPORTE(4)	El host confirma la recepcion del contenido del archivo con un ACK
120	TCP	TRANSPORTE(4)	El host envia una solicitud de cierre de conexión activando la bandera FIN
121	TCP	TRANSPORTE(4)	El proxy envia la confirmacion de la solicitud de cierre activando la bandera ACK y envia su propia solicitud de cierre activando la bandera FIN tambien
127	TCP	TRANSPORTE(4)	El host confirma la solicitud de cierre del proxy activando la bandera ACK

Ejercicio 5.

(Esta en el archivo png “Ejercicio 5 diagrama”)

Ejercicio 6.

En este caso use cómo ejemplo la transferencia del index.html (2da conexión tcp CLIENTE->PROXY)

