

UNIVERSIDAD NACIONAL DE CÓRDOBA
Facultad de Ciencias Exactas, Físicas y Naturales



Redes de Computadoras

Práctico 6: Análisis de tráfico TCP y UDP en GNU/Linux

Contrera, Ivan
Malano, Leandro

Parte 1:

1.1-2)

Con el comando `docker-compose up` (ahora en adelante DC), se crea e inicia el contenedor.

En donde el archivo `Dockerfile` (el cual es cargado por el `docker-compose.yml`), especifica la imagen a usar ("`sameersbn/bind:latest`") y se instala el módulo para que webmin utilice ipv6.

Se setea variable de entorno "`ROOT_PASSWORD=ubuntu`" en el container para usar webmin.

En `docker-compose.yml` se configura la dirección ipv6 para acceder a webmin.

Parte 2:

2.1) Primero cuando se accede a "webmin" se establece una conexión TCP entre host y el contenedor.

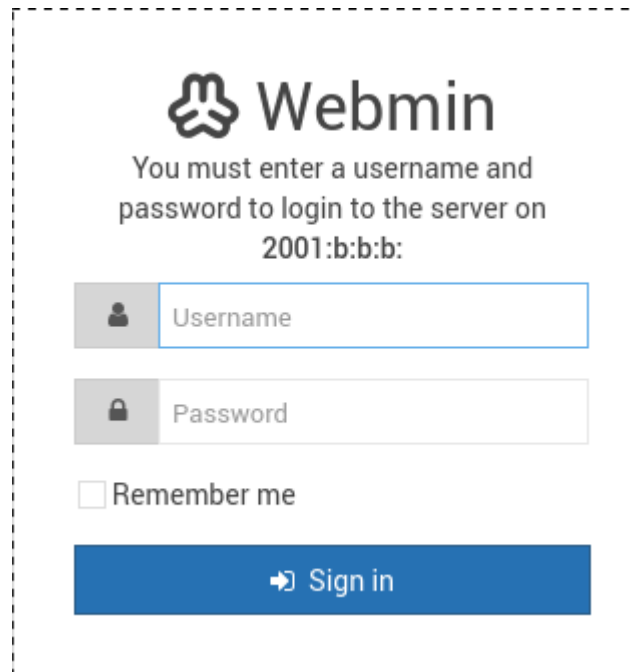
Luego se observan mensajes periódicos (también de TCP) entre ambos, ya que se mantiene información en tiempo real como datos del uso del cpu, memoria, etc.

Apply a display filter ... <Ctrl-/>							Expression...	+
No.	Time	Source	Destination	Protocol	Length	Info		
1	0.000000000	2001:b:b:b::1	2001:b:b:b::2	TCP	94	46456 → 10000 [SYN] Seq=0 Win=28800 Len=0 MSS=1440 SACK_PERM=1 TSval=2300474 TSecr=0 WS=128		
2	0.000063601	2001:b:b:b::2	2001:b:b:b::1	TCP	94	10000 → 46456 [SYN, ACK] Seq=0 Ack=1 Win=28560 Len=0 MSS=1440 SACK_PERM=1 TSval=2300474 TSecr=2300474 WS=1...		
3	0.000108440	2001:b:b:b::1	2001:b:b:b::2	TCP	86	46456 → 10000 [ACK] Seq=1 Ack=1 Win=28800 Len=0 TSval=2300474 TSecr=2300474		
4	0.000249700	2001:b:b:b::1	2001:b:b:b::2	TCP	94	46458 → 10000 [SYN] Seq=0 Win=28800 Len=0 MSS=1440 SACK_PERM=1 TSval=2300474 TSecr=0 WS=128		
5	0.000276807	2001:b:b:b::2	2001:b:b:b::1	TCP	94	10000 → 46458 [SYN, ACK] Seq=0 Ack=1 Win=28560 Len=0 MSS=1440 SACK_PERM=1 TSval=2300474 TSecr=2300474 WS=1...		
6	0.000303053	2001:b:b:b::1	2001:b:b:b::2	TCP	86	46458 → 10000 [ACK] Seq=1 Ack=1 Win=28800 Len=0 TSval=2300474 TSecr=2300474		
7	0.000409911	2001:b:b:b::1	2001:b:b:b::2	TCP	94	46460 → 10000 [SYN] Seq=0 Win=28800 Len=0 MSS=1440 SACK_PERM=1 TSval=2300474 TSecr=0 WS=128		
8	0.000435750	2001:b:b:b::2	2001:b:b:b::1	TCP	94	10000 → 46460 [SYN, ACK] Seq=0 Ack=1 Win=28560 Len=0 MSS=1440 SACK_PERM=1 TSval=2300474 TSecr=2300474 WS=1...		
9	0.000460785	2001:b:b:b::1	2001:b:b:b::2	TCP	86	46460 → 10000 [ACK] Seq=1 Ack=1 Win=28800 Len=0 TSval=2300474 TSecr=2300474		
10	0.000899852	2001:b:b:b::1	2001:b:b:b::2	TCP	603	46456 → 10000 [PSH, ACK] Seq=1 Ack=1 Win=28800 Len=517 TSval=2300474 TSecr=2300474		
11	0.000926232	2001:b:b:b::2	2001:b:b:b::1	TCP	86	10000 → 46456 [ACK] Seq=1 Ack=518 Win=29696 Len=0 TSval=2300474 TSecr=2300474		
12	0.001120743	2001:b:b:b::1	2001:b:b:b::2	TCP	603	46458 → 10000 [PSH, ACK] Seq=1 Ack=1 Win=28800 Len=517 TSval=2300474 TSecr=2300474		
13	0.001143792	2001:b:b:b::2	2001:b:b:b::1	TCP	86	10000 → 46458 [ACK] Seq=1 Ack=518 Win=29696 Len=0 TSval=2300474 TSecr=2300474		
14	0.001368252	2001:b:b:b::1	2001:b:b:b::2	TCP	603	46460 → 10000 [PSH, ACK] Seq=1 Ack=1 Win=28800 Len=517 TSval=2300474 TSecr=2300474		
15	0.001393537	2001:b:b:b::2	2001:b:b:b::1	TCP	86	10000 → 46460 [ACK] Seq=1 Ack=518 Win=29696 Len=0 TSval=2300474 TSecr=2300474		
16	0.003317908	2001:b:b:b::2	2001:b:b:b::1	TCP	223	10000 → 46456 [PSH, ACK] Seq=1 Ack=518 Win=29696 Len=137 TSval=2300475 TSecr=2300474		
17	0.003362231	2001:b:b:b::1	2001:b:b:b::2	TCP	86	46456 → 10000 [ACK] Seq=518 Ack=138 Win=29952 Len=0 TSval=2300475 TSecr=2300475		

2.2) Para acceder a webmin, desde el navegador se ingresa la dirección: [https://\[2001:b:b:b::2\]:10000](https://[2001:b:b:b::2]:10000) , indicando el puerto 10000 siendo el puerto que utiliza la aplicación.

La dirección anterior, es la configurada en el contenedor, en el archivo `docker-compose.yml`

Como pantalla de inicio, se indica que debemos iniciar sesión, cuyo usuario es "root" y su pass es "ubuntu" según se definió en la variable de entorno puesta en `docker-compose.yml`



Webmin

You must enter a username and password to login to the server on
2001:b:b:b:

User

Pass

☐ Remember me

➔ Sign in

2.3) La secuencia de iniciación de una sesión TCP se realiza en 3 pasos: En el primero el cliente (host) envía un mensaje TCP con el bit SYN activo cuyo puerto de destino es 10000.

```

▶ Frame 1: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
▶ Ethernet II, Src: 02:42:63:16:5e:a8 (02:42:63:16:5e:a8), Dst: 02:42:ac:12:00:02 (02:42:ac:12:00:02)
▶ Internet Protocol Version 6, Src: 2001:b:b:b::1, Dst: 2001:b:b:b::2
▼ Transmission Control Protocol, Src Port: 46456, Dst Port: 10000, Seq: 0, Len: 0
  Source Port: 46456
  Destination Port: 10000
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 0
  Header Length: 40 bytes
▼ Flags: 0x002 (SYN)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...0 = Acknowledgment: Not set
  .... .... 0... = Push: Not set
  .... .... .0.. = Reset: Not set
  ▶ .... .... ..1. = Syn: Set
  .... .... ...0 = Fin: Not set
  [TCP Flags: .....S.]
  Window size value: 28800
  [Calculated window size: 28800]
  Checksum: 0x4075 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

```

En el segundo paso, el servidor (contenedor con webmin) responde con un mensaje TCP con los bits SYN y ACK en 1, confirmando que el puerto 10000 esta activo y se puede establecer una conexión.

```

▶ Frame 2: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
▶ Ethernet II, Src: 02:42:ac:12:00:02 (02:42:ac:12:00:02), Dst: 02:42:f8:b7:0d:e0 (02:42:f8:b7:0d:e0)
▶ Internet Protocol Version 6, Src: 2001:b:b:b::2, Dst: 2001:b:b:b::1
▼ Transmission Control Protocol, Src Port: 10000, Dst Port: 40310, Seq: 0, Ack: 1, Len: 0
  Source Port: 10000
  Destination Port: 40310
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 40 bytes
  ▼ Flags: 0x012 (SYN, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... ..... 0... = Reset: Not set
    ▶ .... .... .1. = Syn: Set
    .... .... ...0 = Fin: Not set
    [TCP Flags: .....A.S.]
    Window size value: 28560
    [Calculated window size: 28560]
    [Window size scaling factor: 128]
    Checksum: 0x406d [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    [Urgent pointer: 0]
    [Options: (40 bytes) No-Operation (NOP), No-Operation (NOP), Timestamp

```

Finalmente, en el tercer paso, en caso que se pueda establecer la conexión, el cliente envía un mensaje con el bit ACK en alto.

```

  Destination Port: 10000
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 32 bytes
  ▼ Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... ..... 0... = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
    [TCP Flags: .....A....]
    Window size value: 225
    [Calculated window size: 28800]
    [Window size scaling factor: 128]
    Checksum: 0x406d [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    [Urgent pointer: 0]
    [Options: (40 bytes) No-Operation (NOP), No-Operation (NOP), Timestamp

```

Y a partir de ese momento, se puede comenzar a enviar los datos deseados.

Para el cierre de la conexión, se hace en cuatro fases: En la primera, el que desea cerrar la conexión (contenedor en este caso), envía un segmento TCP con los bits FIN, PSH y ACK en 1, indicando el deseo de la desconexión:

```

▶ Frame 23: 598 bytes on wire (4784 bits), 598 bytes captured (4784 bits) on interface 0
▶ Ethernet II, Src: 02:42:ac:12:00:02 (02:42:ac:12:00:02), Dst: 02:42:f8:b7:0d:e0 (02:42:f8:b7:0d:e0)
▶ Internet Protocol Version 6, Src: 2001:b:b:b::2, Dst: 2001:b:b:b::1
▼ Transmission Control Protocol, Src Port: 10000, Dst Port: 40312, Seq: 198, Ack: 1140, Len: 512
  Source Port: 10000
  Destination Port: 40312
  [Stream index: 1]
  [TCP Segment Len: 512]
  Sequence number: 198 (relative sequence number)
  [Next sequence number: 711 (relative sequence number)]
  Acknowledgment number: 1140 (relative ack number)
  Header Length: 32 bytes
  ▼ Flags: 0x019 (FIN, PSH, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    ....0... = Congestion Window Reduced (CWR): Not set
    ....0... = ECN-Echo: Not set
    ....0... = Urgent: Not set
    ....1... = Acknowledgment: Set
    ....1... = Push: Set
    ....0... = Reset: Not set
    ....0... = Syn: Not set
    ▶ ....1... = Fin: Set
    [TCP Flags: .....AP..F]
  Window size value: 241
  [Calculated window size: 30848]

```

Pero puede pasar que el que no solicitó la desconexión aun quiera seguir enviando datos, y lo podrá hacer, mientras que el otro, ya no.

En la segunda, el receptor del mensaje anterior responde con un ACK indicando que lo recibió correctamente:

```

▶ Frame 24: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▶ Ethernet II, Src: 02:42:f8:b7:0d:e0 (02:42:f8:b7:0d:e0), Dst: 02:42:ac:12:00:02 (02:42:ac:12:00:02)
▶ Internet Protocol Version 6, Src: 2001:b:b:b::1, Dst: 2001:b:b:b::2
▼ Transmission Control Protocol, Src Port: 40312, Dst Port: 10000, Seq: 1140, Ack: 711, Len: 0
  Source Port: 40312
  Destination Port: 10000
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1140 (relative sequence number)
  Acknowledgment number: 711 (relative ack number)
  Header Length: 32 bytes
  ▼ Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    ....0... = Congestion Window Reduced (CWR): Not set
    ....0... = ECN-Echo: Not set
    ....0... = Urgent: Not set
    ....1... = Acknowledgment: Set
    ....0... = Push: Not set
    ....0... = Reset: Not set
    ....0... = Syn: Not set
    ....0... = Fin: Not set
    [TCP Flags: .....A....]
  Window size value: 242
  [Calculated window size: 30976]
  [Window size scaling factor: 128]

```

En la tercera fase, el host va a iniciar por su lado la desconexión, enviando un segmento con el bit FIN en 1:

```

▶ Frame 25: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▶ Ethernet II, Src: 02:42:f8:b7:0d:e0 (02:42:f8:b7:0d:e0), Dst: 02:42:ac:12:00:02 (02:42:ac:12:00:02)
▶ Internet Protocol Version 6, Src: 2001:b:b:b::1, Dst: 2001:b:b:b::2
▼ Transmission Control Protocol, Src Port: 40312, Dst Port: 10000, Seq: 1140, Ack: 711, Len: 0
  Source Port: 40312
  Destination Port: 10000
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1140 (relative sequence number)
  Acknowledgment number: 711 (relative ack number)
  Header Length: 32 bytes
  ▼ Flags: 0x011 (FIN, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    ▶ .... .... ...1 = Fin: Set
    [TCP Flags: .....A...F]
  Window size value: 242
  [Calculated window size: 30976]
  [Window size scaling factor: 128]

```

Y por ultimo, el contenedor envia un mensaje de confirmación ACK:

```

▶ Frame 25: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▶ Ethernet II, Src: 02:42:f8:b7:0d:e0 (02:42:f8:b7:0d:e0), Dst: 02:42:ac:12:00:02 (02:42:ac:12:00:02)
▶ Internet Protocol Version 6, Src: 2001:b:b:b::1, Dst: 2001:b:b:b::2
▼ Transmission Control Protocol, Src Port: 40312, Dst Port: 10000, Seq: 1140, Ack: 711, Len: 0
  Source Port: 40312
  Destination Port: 10000
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1140 (relative sequence number)
  Acknowledgment number: 711 (relative ack number)
  Header Length: 32 bytes
  ▼ Flags: 0x011 (FIN, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    ▶ .... .... ...1 = Fin: Set
    [TCP Flags: .....A...F]
  Window size value: 242
  [Calculated window size: 30976]
  [Window size scaling factor: 128]

```

El tamaño de la ventana (windows size) indica la máxima cantidad de bytes que pueden enviarse al receptor sin esperar un mensaje de confirmación ACK. Este campo está indicado en *window size value*.

Dicho tamaño se calcula con la siguiente ecuación:

$$\text{ventana} = \text{tamañoBuffer} - (\text{ultimoByteRecibido} - \text{ultimoByteLeido})$$

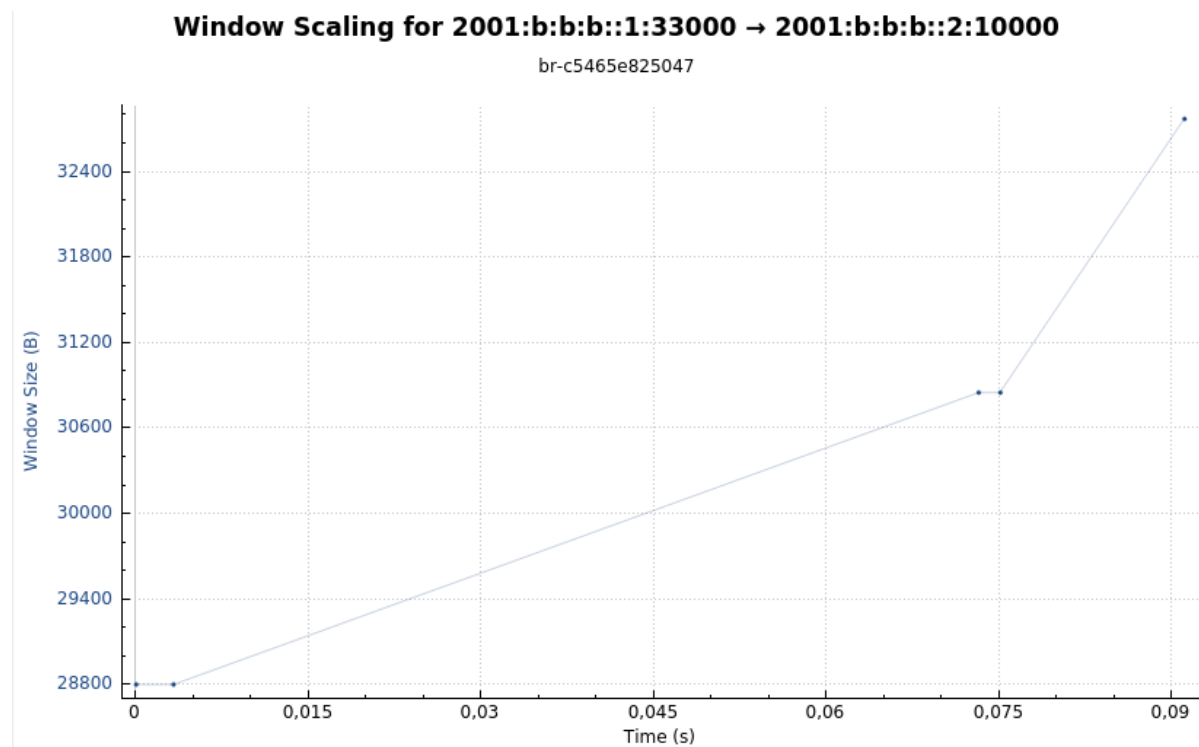
donde, por ejemplo, si el valor indicado es de un segmento enviado por el contenedor y recibido por el host, se le estaría indicando a éste último que puede enviar hasta *ventana* bytes. *tamañoBuffer* hace referencia al tamaño del buffer del emisor; *ultimoByteRecibido*, al último byte que recibió, pero que sigue en el buffer; y *ultimoByteLeido*, al último byte que fue pasado a la capa de aplicación y que ya no ocupa lugar en el buffer. Un ejemplo del valor de la ventana recibido por el host:

31	2.997883315	2001:b:b:b::2	2001:b:b:b::1	TCP	86	10000 → 40314 [ACK] Seq=1 Ack=518 Win=29696 Len=0 TSval=3937570926 TSecr=2204861725
----	-------------	---------------	---------------	-----	----	---

donde en *Win* se indica que se pueden enviar hasta 29696 bytes.

Si la ventana es cero ($Win=0$), significa que el buffer del que envió el segmento actual está lleno y no puede recibir más datos.

Analizando una de las muchas sesiones tcp observadas, utilizando la herramienta *window scaling* se puede observar el tamaño de la ventana del emisor (host, cuyo puerto es 33000):



Comenzando con un tamaño de 28800 bytes y finaliza con un tamaño de 32768 bytes.

3.1) Para que webmin trabaje sin ssl, desde el contenedor de bind se debe editar el archivo de configuración *miniserv.conf* ubicado en */etc/webmin*. Entre todos los parámetros disponibles, se debe cambiar **ssl=1** por **ssl=0**.

3.2)

Se observa un paquete que se envía al contenedor perteneciente a la capa de aplicación (HTTP) de tipo POST, donde se envía tanto el usuario como la contraseña sin encriptar, comprobando la facilidad de poder capturar contraseñas sin utilizar protocolos de seguridad:

```

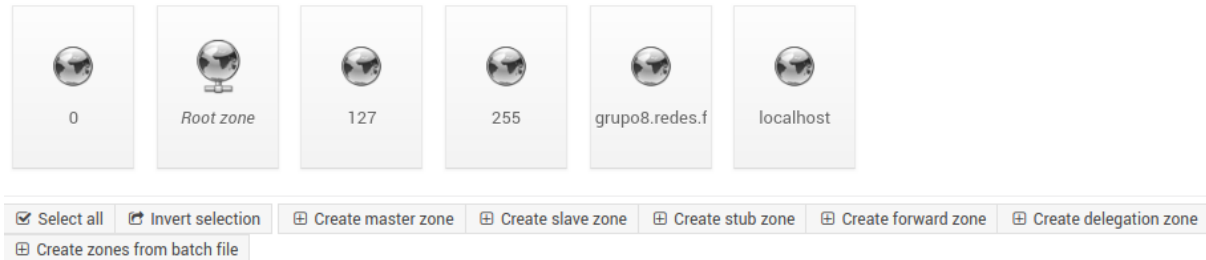
▶ Frame 1: 740 bytes on wire (5920 bits), 740 bytes captured (5920 bits) on interface 0
▶ Ethernet II, Src: 02:42:88:20:f2:e0 (02:42:88:20:f2:e0), Dst: 02:42:ac:12:00:02 (02:42:ac:12:00:02)
▶ Internet Protocol Version 6, Src: 2001:b:b:b::1, Dst: 2001:b:b:b::2
▶ Transmission Control Protocol, Src Port: 36264, Dst Port: 10000, Seq: 1, Ack: 1, Len: 654
▶ Hypertext Transfer Protocol
  ▶ HTML Form URL Encoded: application/x-www-form-urlencoded
    ▶ Form item: "user" = "root"
    ▶ Form item: "pass" = "ubuntu"

```


EJERCICIO 2:

1.1) Para que los datos queden persistentes aunque el contenedor se elimine, se agrega una línea en la sección *volumes* en el archivo *docker-compose.yml*: - /srv/docker/bind:/data

1.2) Se crea una nueva zona maestra desde webmin, cuyo nombre de dominio será grupo8.redes.fcefyn.unc.edu.local:



Una zona maestra carga los datos de zona directamente a partir de un archivo de un sistema principal. Puede contener una subzona o zona hija. También puede contener registros de recursos, por ejemplo registros del sistema principal, alias (CNAME), dirección (AAAA), etc.

1.3) A continuación, se agregan dos registros del tipo AAAA (direcciones ipv6), cuyos nombres serán:

- servidor.grupo8.redes.fcefyn.unc.edu.local
- cliente.grupo8.redes.fcefyn.unc.edu.local

Para hacer esto, se accede al botón de la zona creada (mostrada anteriormente), y luego a otro botón, cuyo nombre es *IPv6 address*:



en la sección **name** se coloca el nombre de una dirección y en **address** la ip correspondiente:

Name Time-To-Live ☒ Default ☐ seconds ▼

Address

Update reverse? ☒ Yes ☐ Yes (and replace existing) ☐ No

Show records matching:

☒ Select all ☐ Invert selection

Name	TTL	IPv6 Address
<input type="checkbox"/> servidor.grupo8.redes.fcefyn.unc.edu.local.	Default	2001:aaaa:aaaa:2::3
<input type="checkbox"/> cliente.grupo8.redes.fcefyn.unc.edu.local.	Default	2001:aaaa:bbbb:4::5

2.1) Utilizando el comando `dig @2001:b:b:b::2 (nombre de dominio) AAAA` se puede acceder a la información mostrada a continuación:

```
root@ivanovic-X555LD:/home/ivanovic/Bind# dig @2001:b:b:b::2 servidor.grupo8.redes.fcefyn.unc.edu.local. AAAA
; <<>> DiG 9.10.3-P4-Ubuntu <<>> @2001:b:b:b::2 servidor.grupo8.redes.fcefyn.unc.edu.local. AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 14609
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;servidor.grupo8.redes.fcefyn.unc.edu.local. IN AAAA

;; ANSWER SECTION:
servidor.grupo8.redes.fcefyn.unc.edu.local. 38400 IN AAAA 2001:aaaa:aaaa:2::3

;; AUTHORITY SECTION:
grupo8.redes.fcefyn.unc.edu.local. 38400 IN NS c6bf0ebd41b7.

;; Query time: 2 msec
;; SERVER: 2001:b:b:b::2#53(2001:b:b:b::2)
;; WHEN: Wed Jun 06 19:41:08 ART 2018
;; MSG SIZE rcvd: 125
```

```
root@ivanovic-X555LD:/home/ivanovic/Bind# dig @2001:b:b:b::2 cliente.grupo8.redes.fcefyn.unc.edu.local. AAAA
; <<>> DiG 9.10.3-P4-Ubuntu <<>> @2001:b:b:b::2 cliente.grupo8.redes.fcefyn.unc.edu.local. AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 40713
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;cliente.grupo8.redes.fcefyn.unc.edu.local. IN AAAA

;; ANSWER SECTION:
cliente.grupo8.redes.fcefyn.unc.edu.local. 38400 IN AAAA 2001:aaaa:bbbb:4::5

;; AUTHORITY SECTION:
grupo8.redes.fcefyn.unc.edu.local. 38400 IN NS c6bf0ebd41b7.

;; Query time: 0 msec
;; SERVER: 2001:b:b:b::2#53(2001:b:b:b::2)
;; WHEN: Wed Jun 06 19:41:24 ART 2018
;; MSG SIZE rcvd: 124
```

se puede observar que en la sección **ANSWER SECTION** se muestra la resolución de la ip buscada.

2.2) Los campos de cabecera UDP son:

Puerto de origen: 37684

Puerto de destino: 53

Longitud del segmento: 78 bytes

Suma de comprobación

▼ User Datagram Protocol, Src Port: 37684, Dst Port: 53

Source Port: 37684

Destination Port: 53

Length: 78

Checksum: 0x40a6 [unverified]

[Checksum Status: Unverified]

[Stream index: 0]

► Domain Name System (query)

La información del protocolo DNS está por encima de la capa de transporte, y es la siguiente (consulta):

▼ Domain Name System (query)

[Response In: 29]

Transaction ID: 0xb178

▼ Flags: 0x0120 Standard query

0... .. = Response: Message is a query

.000 0... .. = Opcode: Standard query (0)

... ..0. = Truncated: Message is not truncated

... ..1 = Recursion desired: Do query recursively

... ..0.. = Z: reserved (0)

... ..1. = AD bit: Set

... ..0 = Non-authenticated data: Unacceptable

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 1

► Queries

► Additional records

Se pueden destacar algunos campos: Dentro de *flags*, cada bit es una bandera. En la primera línea, el flag indica si el mensaje es de consulta (0) o de respuesta (1). El campo *Opcode* de 4 bits especifica el tipo de consulta: Consulta estándar (1), consulta inversa (1) o solicitud del estado del servidor (2). El resto de los valores se reservan para su uso futuro. *Recursion desired*, que indica si la solicitud va a ser recursiva (1) o no (0).

En la respuesta se observa lo siguiente:

```

▼ Domain Name System (response)
  [Request In: 28]
  [Time: 0.000180649 seconds]
  Transaction ID: 0xb178
  ▼ Flags: 0x8580 Standard query response, No error
    1... .. = Response: Message is a response
    .000 0... .. = Opcode: Standard query (0)
    .... 1... .. = Authoritative: Server is an authority for domain
    .... 0... .. = Truncated: Message is not truncated
    .... 1... .. = Recursion desired: Do query recursively
    .... 1... .. = Recursion available: Server can do recursive queries
    .... 0... .. = Z: reserved (0)
    .... 0... .. = Answer authenticated: Answer/authority portion was not authenticated by the server
    .... 0... .. = Non-authenticated data: Unacceptable
    .... 0000 = Reply code: No error (0)
  Questions: 1
  Answer RRs: 1
  Authority RRs: 1
  Additional RRs: 1
  ► Queries
  ▼ Answers
    ► cliente.grupo8.redes.fcefyn.unc.edu.local: type AAAA, class IN, addr 2001:aaaa:bbbb:4::5
    ► Authoritative nameservers
    ► Additional records

```

ES muy similar al de consulta, pero se agrega un campo importante: *Answers*, el cual trae la respuesta que se solicitó. Indica que la información es de tipo AAAA (direcciones ipv6), cuya dirección es 2001:aaaa:bbbb:4::5, y el nombre de dominio que se quiso consultar (cliente.grupo8.redes.fcefyn.unc.edu.local).

Una **zona de autoridad** es una parte del espacio de nombre de dominios sobre la que es responsable un servidor DNS, que puede tener autoridad sobre varias zonas. Es este trabajo, la zona (o servidor) autoritaria es *grupo8.redes.fcefyn.unc.edu.local*:

```

▼ Answers
  ► cliente.grupo8.redes.fcefyn.unc.edu.local: type AAAA, class IN, addr 2001:aaaa:bbbb:4::5
  ▼ Authoritative nameservers
    ► grupo8.redes.fcefyn.unc.edu.local: type NS, class IN, ns c6bf0ebd41b7
    ► Additional records

```

Como no es un protocolo orientado a la conexión, si el segmento no llega a destino, el mismo no lo volverá a enviar (ni tampoco se enterará). Al no poder resolverse la solicitud, la aplicación solicitante (navegador web, servidor ftp, etc) deberá hacer algo para manejar el problema.

ENLACES:

<https://faq.active24.com/es/662420-Configuraci%C3%B3n-de-registros-DNS-A-AAAA-CNAME-MX-TXT->

<http://www.damagehead.com/blog/2015/04/28/deploying-a-dns-server-using-docker/>

https://es.wikipedia.org/wiki/Sistema_de_nombres_de_dominio

<http://www.newdevices.com/tutoriales/dns/3.html>