

Informe final laboratorio N3

Creación de paquetes

Integrantes:

- José Carvajal
- Agustin Moore
- Diego Torreblanca
- Enzo Urrutia

Profesor:

- Jaime Álvarez

Ayudante:

- Alexis Inzunza

Fecha: 13 de Abril del 2016

Santiago

Indice

- Pasos previos al envío de paquetes..... Página 3
- Caso 1: Envío a FF:FF:FF:FF:FF:FF..... Página 4
- Caso 2: Envío a MAC de otro equipo..... Página 5
- Caso 3: Envío a MAC de ningun equipo de la red..... Página 6
- Conclusión..... Página 7

Pasos previos al envío de paquetes

Antes de definir el punto al cual deseamos enviar nuestro paquete de datos, debemos realizar pasos para poder apuntar al punto donde queremos. Para esto, utilizamos la herramienta Scapy, la cual empleamos desde la consola del terminal. Este programa nos permite manipular los paquetes de datos, primero creándolos, para luego enviar en o las direcciones que se desee.

- Primeras instrucciones dentro de Scapy

```
telematica@Telematica03: ~  
telematica@Telematica03:~$ sudo scapy  
[sudo] password for telematica:  
INFO: Can't import python gnuplot wrapper . Won't be able to plot.  
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().  
WARNING: No route found for IPv6 destination :: (no default route?)  
Welcome to Scapy (2.2.0)  
>>> Enlace=Ether()  
>>> ls(Enlace)  
WARNING: Mac address to reach destination not found. Using broadcast.  
dst      : DestMACField      = 'ff:ff:ff:ff:ff:ff' (None)  
src      : SourceMACField    = '00:00:00:00:00:00' (None)  
type     : XShortEnumField   = 0              (0)
```

- Información de la red, según la dirección elegida (Ejemplo)

```
>>> red=IP()  
>>> ls(red)  
version   : BitField          = 4              (4)  
ihl       : BitField          = None           (None)  
tos       : XByteField        = 0              (0)  
len       : ShortField        = None         (None)  
id        : ShortField        = 1              (1)  
flags     : FlagsField        = 0              (0)  
frag      : BitField          = 0              (0)  
ttl       : ByteField         = 64             (64)  
proto     : ByteEnumField     = 0              (0)  
chksum    : XShortField       = None           (None)  
src       : Emph              = '127.0.0.1'  (None)  
dst       : Emph              = '127.0.0.1'  ('127.0.0.1')  
options   : PacketListField  = []           ([])  
>>> Transporte=ICMP()  
>>> ls(Transporte)  
type      : ByteEnumField     = 8              (8)  
code      : MultiEnumField    = 0              (0)  
chksum    : XShortField       = None           (None)  
id        : ConditionalField  = 0              (0)  
seq       : ConditionalField  = 0              (0)  
ts_ori    : ConditionalField  = 62555674      (62555674)  
ts_rx     : ConditionalField  = 62555674      (62555674)  
ts_tx     : ConditionalField  = 62555674      (62555674)  
gw        : ConditionalField  = '0.0.0.0'     ('0.0.0.0')  
ptr       : ConditionalField  = 0              (0)  
reserved  : ConditionalField  = 0              (0)  
addr_mask : ConditionalField  = '0.0.0.0'     ('0.0.0.0')
```

Los pasos mencionados anteriormente, solo fueron los que debimos realizar antes de establecer el punto donde se querían enviar él, o los paquetes de datos. A continuación, los casos según los puntos de envío elegidos.

Caso 1: Envío a FF:FF:FF:FF:FF:FF

El primer caso en este análisis, es en cual se busca enviar con una dirección "FF", desde este punto es que surgen las interrogantes respecto a lo que sucede bajo estas circunstancias.

Para poder obtener los datos estudiados en este caso, debimos apuntar en la dirección a la cual se deseaba enviar, como lo muestra la siguiente imagen. En este paso, es donde debíamos seleccionar el punto a donde enviar el paquete de datos, en este caso, fue hacia una dirección FF:FF:FF:FF:FF:FF.

```
>>> EnterSrc: 18:a9:05:1e:fb:42
>>> ls(Enlace)
dst      : DestMACField      = 'FF:FF:FF:FF:FF:FF' (None)
src      : SourceMACField    = '18:a9:05:1e:fb:42' (None)
.type    : XShortEnumField   = 0
>>> red=TP()
```

¿Qué pasa cuando envió un paquete a la dirección FF:FF:FF:FF:FF:FF?

¿Quiénes lo reciben? ¿Por qué?

Cuando se envía un paquete a la dirección FF:FF:FF:FF:FF:FF, la cual corresponde a una dirección broadcast, por lo cual todos los dispositivos reciben el paquete de datos en el caso de que sea un hub, ya que estos hacen pasar los paquetes por toda la red. En el caso de un switch no ocurriría esto, ya que solo se puede enviar a direcciones concretas.

La siguiente imagen, demuestra como uno de los equipos conectados a esta red, recibe el paquete de datos enviado, sin que haya sido el, quien debería recibirlo únicamente.

| Filter: icmp | | | | | | | Expression... | Clear | Apply | Save |
|--------------|--------------|-----------|-------------|----------|--------|--|---------------|-------|-------|------|
| No. | Time | Source | Destination | Protocol | Length | Info | | | | |
| 58292 | 238.16656906 | 127.0.0.1 | 127.0.0.1 | ICMP | 42 | Echo (ping) request id=0x0000, seq=0/0, ttl=64 | | | | |
| 106916 | 720.49456006 | 127.0.0.1 | 127.0.0.1 | ICMP | 42 | Echo (ping) request id=0x0000, seq=0/0, ttl=64 | | | | |
| 107400 | 1076.0026586 | 127.0.0.1 | 127.0.0.1 | ICMP | 42 | Echo (ping) request id=0x0000, seq=0/0, ttl=64 | | | | |

```
Frame 107400: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: Hewlett-1e:fb:42 (18:a9:05:1e:fb:42), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
Internet Control Message Protocol
```

Caso 2: Envío a MAC de otro equipo

Este es el caso, en el cual, al momento de enviar el paquete de datos, se hace en una dirección determinada, a un equipo en particular. El siguiente ejemplo es el que demuestra que paso con este caso, a la hora de realizar el ejercicio. En la siguiente imagen, nosotros apuntamos a una dirección en concreto, a un dispositivo determinado.

```
>>> Enlace.dst="18:a9:05:1e:f6:37"  
>>> paquete=Enlace/red/Transporte/payload  
>>> sendp(paquete)
```

¿Qué pasa cuando envío un paquete a una MAC de otro equipo?

¿Quiénes lo pueden recibir? ¿Por qué? Para este caso lo recibe solo el equipo que cuente la mac ingresada ya que no pueden haber dos mac iguales, porque esta es el sello del hardware y no de la máquina virtual.

```
>>> paquete=Enlace/red/Transporte/payload  
>>> sendp(paquete)
```

Como se puede apreciar en la siguiente imagen, el equipo al cual se le enviaba la información, lo recibe clara y únicamente.

| Filter: icmp | | | | | | Expression... | Clear | Apply | Save |
|--------------|--------------|-----------|-------------|----------|--------|--|-------|-------|------|
| No. | Time | Source | Destination | Protocol | Length | Info | | | |
| 2053 | 281.88779108 | 127.0.0.1 | 127.0.0.1 | ICMP | 60 | Echo (ping) request id=0x0000, seq=0/0, ttl=64 | | | |

```
Frame 2053: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
Ethernet II, Src: Hewlett-1e:fb:42 (18:a9:05:1e:fb:42), Dst: Hewlett-1e:f6:37 (18:a9:05:1e:f6:37)  
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)  
Internet Control Message Protocol
```

Caso 3: Envío a MAC de ningún equipo de la red

¿Qué sucede si envía un paquete a una MAC que no corresponda a ningún equipo de la red? ¿Quiénes lo pueden recepcionar? ¿Por qué?

En una red que cuente con un hub todos los equipos los recepcionarían ya que en este todos los paquetes pasan por el circuito de equipos entero de la red pero ni uno podría recibirlo ya que no se podría autenticar la mac, en un switch no se enviaría a nadie y no podría ser recepcionado por nadie ya que no encontraría una ruta a esta dirección errónea.

```
>>> Enlace.dst="A1:B2:C3:D4:F5:G6"
>>> paquete=Enlace/red/Transporte/payload
>>> sendp(paquete)
Traceback (most recent call last):
  File "<console>", line 1, in <module>
    File "/usr/lib/python2.7/dist-packages/scapy/sendrecv.py", line 259, in sendp
      __gen_send(conf.L2socket(iface=iface, *args, **kargs), x, inter=inter, loop=
loop, count=count, verbose=verbose, realtime=realtime)
    File "/usr/lib/python2.7/dist-packages/scapy/sendrecv.py", line 234, in __gen_
send
      s.send(p)
    File "/usr/lib/python2.7/dist-packages/scapy/supersocket.py", line 32, in send
      sx = str(x)
    File "/usr/lib/python2.7/dist-packages/scapy/packet.py", line 261, in __str__
      return self.build()
    File "/usr/lib/python2.7/dist-packages/scapy/packet.py", line 319, in build
      p = self.do_build()
    File "/usr/lib/python2.7/dist-packages/scapy/packet.py", line 308, in do_build
      pkt = self.self_build()
    File "/usr/lib/python2.7/dist-packages/scapy/packet.py", line 299, in self_bui
ld
      p = f.addfield(self, p, val)
    File "/usr/lib/python2.7/dist-packages/scapy/fields.py", line 70, in addfield
      return s+struct.pack(self.fmt, self.i2m(pkt,val))
    File "/usr/lib/python2.7/dist-packages/scapy/layers/l2.py", line 94, in i2m
      return MACField.i2m(self, pkt, self.i2h(pkt, x))
    File "/usr/lib/python2.7/dist-packages/scapy/fields.py", line 183, in i2m
      return mac2str(x)
    File "/usr/lib/python2.7/dist-packages/scapy/utils.py", line 244, in mac2str
      return "".join(map(lambda x: chr(int(x,16)), mac.split(":")))
    File "/usr/lib/python2.7/dist-packages/scapy/utils.py", line 244, in <lambda>
      return "".join(map(lambda x: chr(int(x,16)), mac.split(":")))
ValueError: invalid literal for int() with base 16: 'G6'
>>> ls(paquete)
dst      : DestMACField      = 'A1:B2:C3:D4:F5:G6' (None)
src      : SourceMACField   = '18:a9:05:1e:fb:42' (None)
type     : XShortEnumField  = 2048          (0)
..
```

Conclusión

Durante la realización de este laboratorio, descubrimos los distintos casos que pueden ocurrir ante el envío de datos, y como puede variar el transporte de estos, según el dispositivo que este instalado en la red, ya sea el switch o hub.

El emplear los programas utilizados durante el laboratorio, nos ayuda a poder descubrir como es la interacción entre los dispositivos conectados a la misma red, ya que en estos software quedaba explícita dicha acción.

Como resultado de todo este laboratorio, podemos concluir, en la importancia que tiene el mecanismo que se instala para la distribución de datos, porque cada uno interactúa de diferente forma con los dispositivos conectados a la red. En el caso del switch, este envía según la dirección señalada, a diferencia del hub, el cual manda el paquete de datos y es recibido por el que presenta la dirección señalada.