



Hub/Switch

PRÁCTICA 4: HUB Y SWITCH

VÍCTOR PÉREZ PEDROSA

ÍNDICE

<u>Ejercicio 1</u>	<u>3</u>
<u>Ejercicio 2</u>	<u>4</u>
<u>Ejercicio 3</u>	<u>5</u>

ÍNDICE DE FIGURAS

Ilustración 1 Velocidad de transmisión de un switch	3
Ilustración 2 Velocidad de transmisión de un HUB	3
Ilustración 3 Comportamiento de un HUB	4
Ilustración 4 Parámetros de Macof	5
Ilustración 5 Opciones de Macof	6
Ilustración 6 ifdown	7
Ilustración 7 Ataque Macof	7
Ilustración 8 Paquetes ICMP	7

VÍDEO: <https://youtu.be/bLv7ioRqi8o>

BLOG: <https://switchv.blogspot.com/2019/01/hub-y-switch.html>

EJERCICIO 1

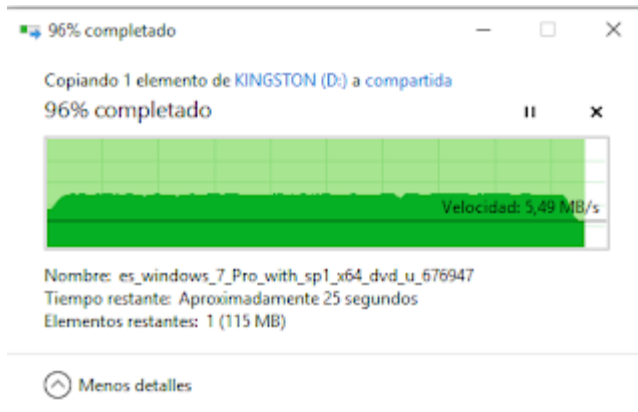


Ilustración 1 Velocidad de transmisión de un switch

En esta imagen podemos observar la velocidad de transmisión de un switch, elemento que hemos utilizado es un router por ello la velocidad de transmisión es baja.

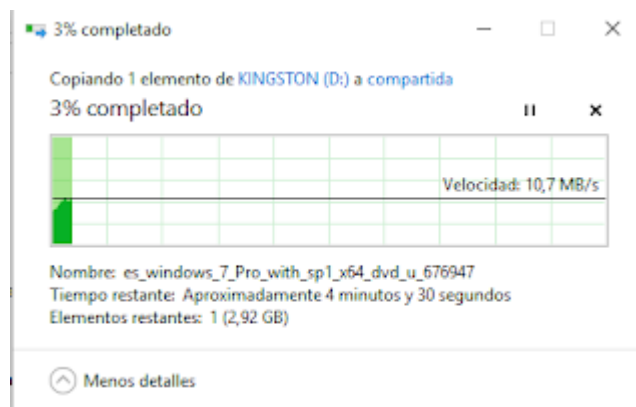


Ilustración 2 Velocidad de transmisión de un HUB

Creamos una carpeta compartida en la cual compartimos un archivo grande para observar la velocidad de transmisión de un switch.

La velocidad es inferior a la velocidad teórica, esto puede depender por el tipo de cable, el hub, etc.

EJERCICIO 2

[illegible]

Ilustración 3 Comportamiento de un HUB

Para realizar este ejercicio vamos a necesitar de una aplicación llamada whreshark en la cual nos va a mostrar el tráfico de datos.

Todos los host se encuentran conectados mediante un HUB y están conectados en la misma red, para comprobar que un HUB envía la información por todas las bocas excepto por la que se lo envía vamos a mandarnos ping entre nosotros y uno de nosotros observa el tráfico de datos, donde observamos que el HUB envía la información por todas sus bocas.

EJERCICIO 3

```
OPTIONS
-i interface
    Specify the interface to send on.

-s src Specify source IP address.

-d dst Specify destination IP address.

-e tha Specify target hardware address.

-x sport
    Specify TCP source port.

-y dport
    Specify TCP destination port.

-n times
    Specify the number of packets to send.
```

Ilustración 4 Parámetros de Macof

Ahora vamos a explicar los parámetros de MACOF.

- -i Sirve para especificar la interfaz por donde va a enviar las tramas.
- -s Para especificar la dirección IP origen
- -d Sirve para especificar la dirección IP de destino
- -e Sirve para especificar la dirección MAC del host.
- -x Especificar el puerto origen TCP
- -y Especificar el puerto de destino TCP
- -n Especificar el número de paquetes que deseamos enviar.

INUNDACIÓN SIMPLE.

Envía tramas aleatorias llenando la tabla CAM del switch, lo cual hace que se comporte como un HUB, lo cual hace que se pueda manipular el tráfico en red.

INUNDACIÓN DIFERIDA.

Sirve para verificar si el switch está sobrecargado

CONTRA MEDIDAS

Limitar el número de direcciones MAC que se conectan al switch.

Filtrado de MAC, limita el número de direcciones MAC.

ATAQUE CON MACOF

Lo primero que hay que hacer en el terminal de kali es ejecutar el coman man-macof en el cual nos muestra una ayuda de los comandos a ejecutar de macof.

```
NAME
  macof - flood a switched LAN with random MAC addresses

SYNOPSIS
  macof [-i interface] [-s src] [-d dst] [-e ethal] [-x sport] [-y dport]
        [-n times]

DESCRIPTION
  macof floods the local network with random MAC addresses (causing some
  switches to fail open in repeating mode, facilitating sniffing). A
  straight C port of the original Perl Net::RawIP macof program by Ian
  Vitek <ian.vitek@infosec.se>.

OPTIONS
  -i interface
      Specify the interface to send on.
  -s src
      Specify source IP address.
  -d dst
      Specify destination IP address.
```

Ilustración 5 Opciones de Macof

Ahora escribimos en el terminal el comando:

“macof -i eth0 -d 192.168.1.1”

Después ejecutaremos los comandos if down y el comando if config

```

root@kali:~# ifdown eth0
RTNETLINK answers: No such process
RTNETLINK answers: Cannot assign requested address
root@kali:~# ifconfig eth0
eth0: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 00:0c:29:b1:33:5d txqueuelen 1000 (Ethernet)
    RX packets 74 bytes 5676 (5.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 39 bytes 3662 (3.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisio

```

Ilustración 6 ifdown

Ahora en la siguiente imagen podremos observar el ataque macof

```

831800945(0) win 512
0:d:29:b1:33:5d b8:87:49:72:60:1d 0.0.0.0.36953 > 0.0.0.0.3436: S 1237199584:1237199584(0) win 512
0:81:48:47:5b:ef b:13:3:4b:b7:df 0.0.0.0.6494 > 0.0.0.0.6856: S 971699787:971699787(0) win 512
b:27:a4:4:14:95 19:af:33:36:c:25 0.0.0.0.19462 > 0.0.0.0.44563: S 1037960821:1037960821(0) win 512
9:a:75:38:1c:8b 53:4b:62:c:20:b5 0.0.0.0.20133 > 0.0.0.0.47318: S 1577464837:1577464837(0) win 512
2:4b:b4:45:e3:da 74:e:2b:24:35:64 0.0.0.0.6432 > 0.0.0.0.21500: S 1258361576:1258361576(0) win 512
f:b0:d6:13:c9:00 7c:bree:52:d2:b2 0.0.0.0.3492 > 0.0.0.0.31451: S 595721977:595721977(0) win 512
9:19:5c:2b:9a:5d 7f:27:94:35:4a:9c 0.0.0.0.8426 > 0.0.0.0.57695: S 70826541:70826541(0) win 512
:28:a0:7c:8d:44 9a:c9:3e:44:42:57 0.0.0.0.39523 > 0.0.0.0.16866: S 1361202014:1361202014(0) win 512
c:18:ed:72:6:6a 56:8d:c0:74:a3:69 0.0.0.0.56843 > 0.0.0.0.43644: S 1989845758:1989845758(0) win 512
4:8e:3f:6e:5c:f0 69:87:b7:56:86:d9 0.0.0.0.24343 > 0.0.0.0.11666: S 1647898626:1647898626(0) win 512
9:2b:be:34:de:76 7e:9f:8b:42:60:0e 0.0.0.0.14946 > 0.0.0.0.59524: S 654517676:654517676(0) win 512

```

Ilustración 7 Ataque Macof

Time	Source	Destination	Protocol	Length	Info
262000	298.311112	1.8.8.3	1.8.8.5	ICMP	74 Echo (ping) request 10-b00001, seq=276/3585, ttl=128 (reply in 262001)
262001	298.311117	1.8.8.5	1.8.8.3	ICMP	74 Echo (ping) reply 10-b00001, seq=276/3585, ttl=128 (request in 262000)
262012	298.321040	1.8.8.2	1.8.8.0	ICMP	74 Echo (ping) request 10-b00001, seq=275/3585, ttl=128 (reply in 262013)
262012	298.321117	1.8.8.3	1.8.8.2	ICMP	74 Echo (ping) reply 10-b00001, seq=275/3585, ttl=128 (request in 262011)
262020	588.842858	Pujitsu:QuantaCo	ARP	42	Who has 1.8.8.3? Tell 1.8.8.5
262058	588.842858	QuantaCo:Pujitsu	ARP	42	1.8.8.3 is at 84:7d:7b:00:5d:05
262444	312.420431	1.8.8.5	1.8.8.5	ICMP	74 Echo (ping) request 10-b00001, seq=272/4895, ttl=128 (reply in 262445)
262445	312.420436	1.8.8.5	1.8.8.3	ICMP	74 Echo (ping) reply 10-b00001, seq=272/4895, ttl=128 (request in 262444)
262802	312.442821	1.8.8.2	1.8.8.0	ICMP	74 Echo (ping) request 10-b00001, seq=273/4703, ttl=128 (reply in 262803)
262802	312.442970	1.8.8.5	1.8.8.2	ICMP	74 Echo (ping) reply 10-b00001, seq=273/4703, ttl=128 (request in 262801)
262747	534.448728	1.8.8.2	1.8.8.0	ICMP	74 Echo (ping) request 10-b00001, seq=274/4600, ttl=128 (reply in 262748)
262748	534.448681	1.8.8.5	1.8.8.2	ICMP	74 Echo (ping) reply 10-b00001, seq=274/4600, ttl=128 (request in 262747)
262388	315.455071	1.8.8.2	1.8.8.5	ICMP	74 Echo (ping) request 10-b00001, seq=275/4865, ttl=128 (reply in 262389)
262388	315.455161	1.8.8.5	1.8.8.2	ICMP	74 Echo (ping) reply 10-b00001, seq=275/4865, ttl=128 (request in 262388)

Ilustración 8 Paquetes ICMP

En esta imagen se ve que envía los ping la máquina 1.0.0.5 y la 1.0.0.2 y las máquinas que estábamos enviando los ping eran la 1.0.0.1 y la 1.0.0.4.

CONCLUSIONES

Para atacar al switch nos costó mucho porque no nos podíamos conectar pero al cambiar de switch ya pudimos hacer el ejercicio.

Esta práctica me ha enseñado mucho porque hemos aprendido a hacer conexión con un switch y con un hub.