Surveillance, port security, snooping et VLAN

Introduction

Nous avons vécu une première attaque par usurpation d'adresse et empoisonnement des caches de nos voisins, le tout sans être vu. Nous avons également vu comment se faire passer pour le serveur DHCP et la passerelle par défaut pour intercepter les paquets à destination d'autres réseaux.
Nous avons évoqué la possibilité de saturer la table des commutateurs et accaparer toutes les adresses disponibles d'un serveur DHCP pour rendre le réseau inopérable.

Dans ce chapitre nous allons étudier les solutions à notre disposition pour sécuriser la couche liaison de données.

Port security

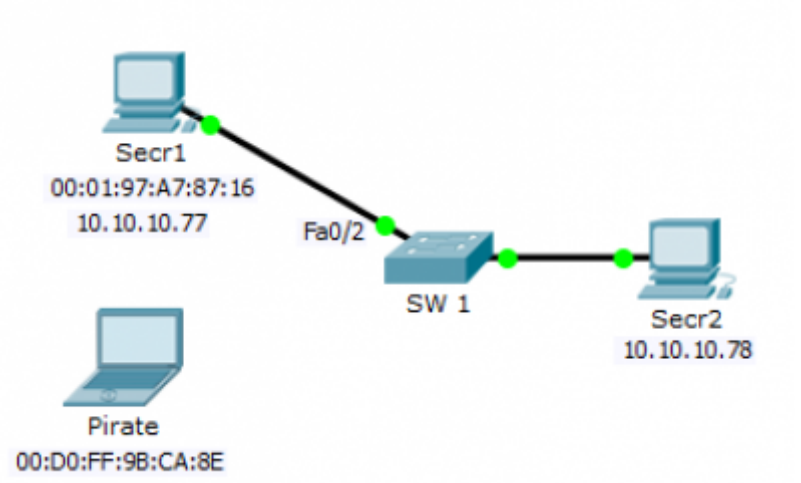
Port security offre au commutateur un contrôle des interfaces connectées à ses ports grâce à leur adresse MAC.

Fonctionnement

Le commutateur contient une table mettant en relation ses ports et les adresses MAC des interfaces qui y sont connectées. Si je débranche une interface et en branche une autre sur un port, l'adresse MAC est mise à jour dans la table du commutateur.

Port security nous permet de choisir une (voire plusieurs) adresse MAC autorisée à se connecter à un port et d'empêcher le passage de trames si une interface non autorisée se connecte à ce même port, ce qui fut le cas lorsque le pirate a branché son ordinateur portable dans un des bureaux de la direction de Corporate.

Situation initiale



Le PC `Secr1` dialogue avec le PC `Secr2` dans le bureau du secrétariat de Corporate, les deux ordinateurs sont sur le même réseau et communiquent grâce au commutateur Cisco 2960 nommé `SW1`. L'interface réseau de `Secr1` ayant pour adresse MAC `00:01:97:A7:87:16` est connectée sur le port Fa0/2 du commutateur.

Configuration de port security

Nous allons protéger le port fa0/2 de telle manière qu'il n'accepte pas d'autre interfaces que celle du PC Secr1 :

```
SW1(config)#interface fa0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport port-security
```

Nous avons activé port security sur l'inteface fa0/2 et nous souhaitons maintenant déclarer l'adresse MAC autorisée:

```
SW1(config-if)#switchport port-security mac-address ?
H.H.H   48 bit mac address
sticky  Configure dynamic secure addresses as sticky
```

Ici deux possibilités : Soit nous déclarons l'adresse MAC de Secr1, soit nous choisisson l'option `sticky` : Cette option permet d'enregister l'adresse de la première interface active sur le port sans avoir à la déclarer manuellement.

```
SW1(config-if)#switchport port-security mac-address sticky
```

Regardons maintenant la configuration de port security pour le port fa0/2 de notre commutateur :

```
SW1#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
               (Count)       (Count)        (Count)
---------------------------------------------------------------------
        Fa0/2        1          1                0         Shutdown
```

```
--------------------------------------------------------------------------
```

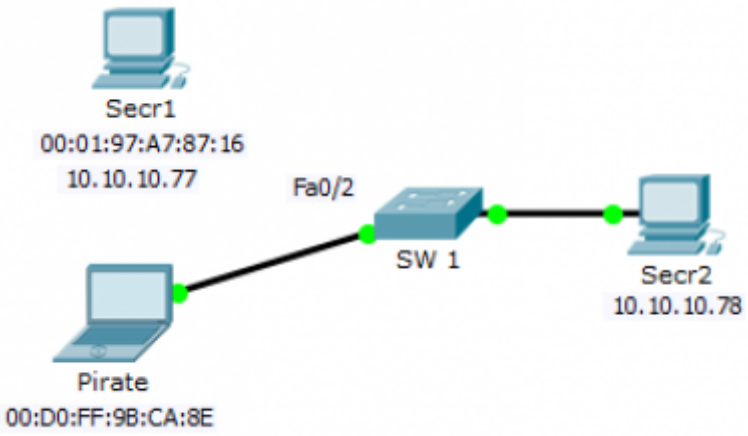
Une adresse MAC est bien enregistrée, le maximum d'adresse possible est à 1 et aucune violation de sécurité n'a eu lieu.

On peut naturellement modifier ces options :

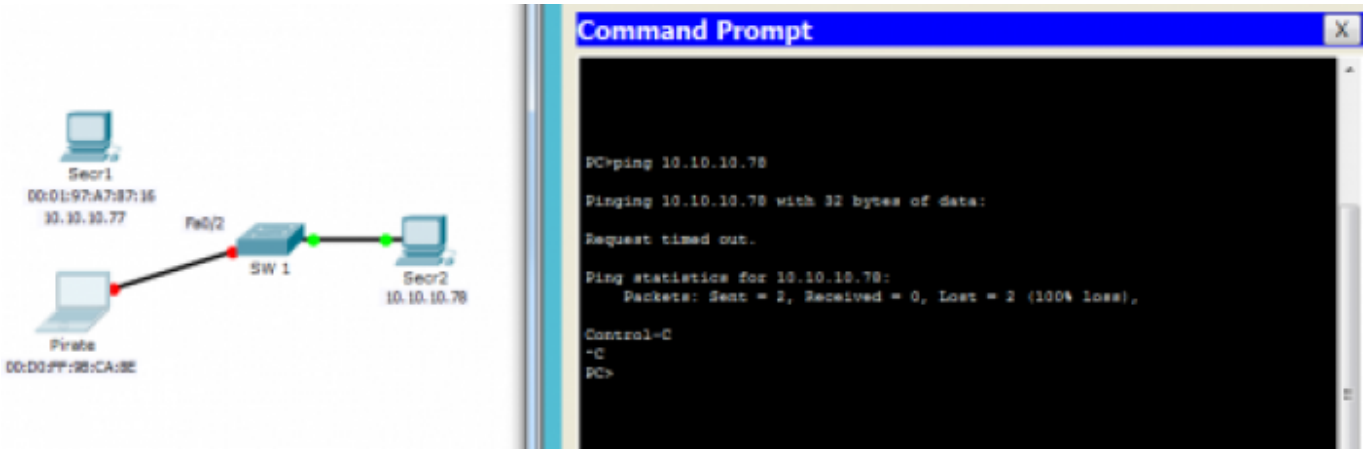
- Modifier le nombre d'adresses MAC autorisées : `switchport port-security maximum X` *(0<X<133)*
- Modifier l'action par défaut en cas de violation de la politique appliquée :
  - `switchport port-security violation protect` : interdiction pure et simple de toute adresse MAC inconnue;
  - `switchport port-security violation shutdown` : le mode par défaut, on désactive l'interface;
  - `switchport port-security violation restrict` : alarme SNMP envoyée sur le réseau (incluant l'interdiction du mode protect).

Démonstration

Maintenant que nous avons configuré port security, tentons de connecter notre ordinateur “pirate” sur le port fa0/2:



Tout se passe bien jusqu'à ce que l'ordinateur pirate tente d'envoyer une trame sur le réseau :



le commutateur désactive immédiatement le port :

```
%LINK-5-CHANGED: Interface FastEthernet0/2, changed state to
administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed
state to down
```

Le port restera désactivé jusqu'à ce que l'administrateur le réactive manuellement (même si on branche l'interface autorisée au départ). En interrogeant la configuration du commutateur on voit qu'une violation d'accès a eu lieu :

```
SW1#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
               (Count)       (Count)        (Count)
---------------------------------------------------------------------
        Fa0/2        1          1                1         Shutdown
---------------------------------------------------------------------
```

Pour réactiver le port : selectionner l'interface (`int fa0/2` dans la cas présent) puis `shutdown` suivi de `no shutdown`: la seule commande `no shutdown` ne fonctionnera pas.

Conclusion

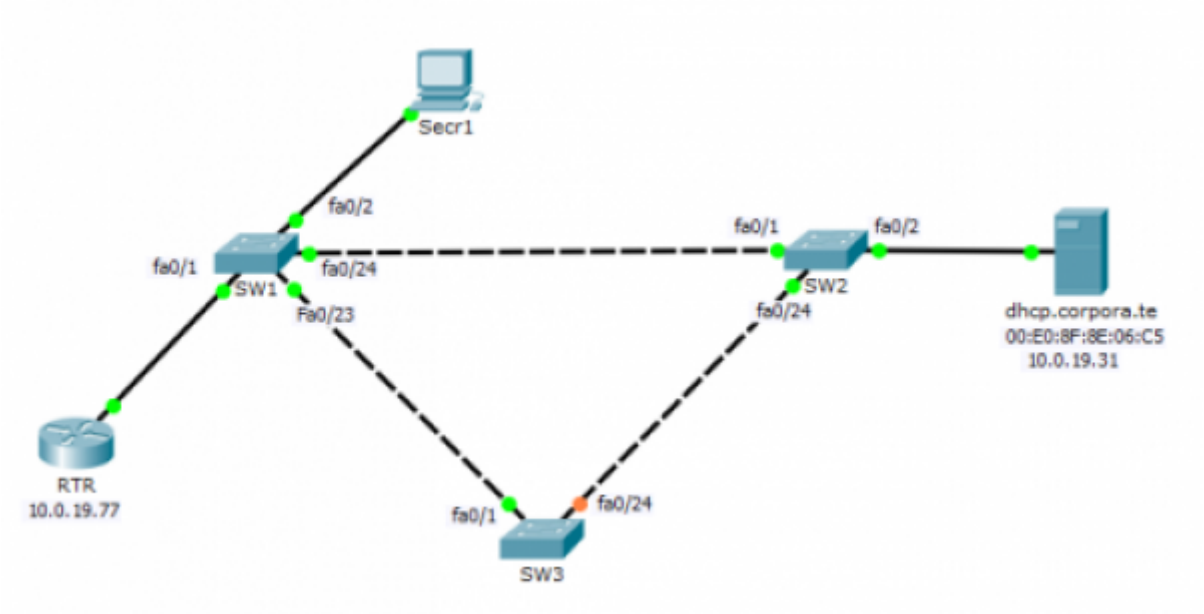
Port security est une solution idéale lorsque l'on souhaite “figer” une configuration et empêcher quelqu'un d'accaparer toutes les adresses disponibles d'un serveur DHCP mais plutôt contraignante si on souhaite connecter des PC nomades par exemple.
Il faut alors faire l'impasse sur une partie de la sécurité (augmenter le nombre d'adresses autorisées à se connecter par port) ou fournir un travail considérable de reconfiguration à chaque changement de connexion, ce qui finira par lasser.

DHCP Snooping

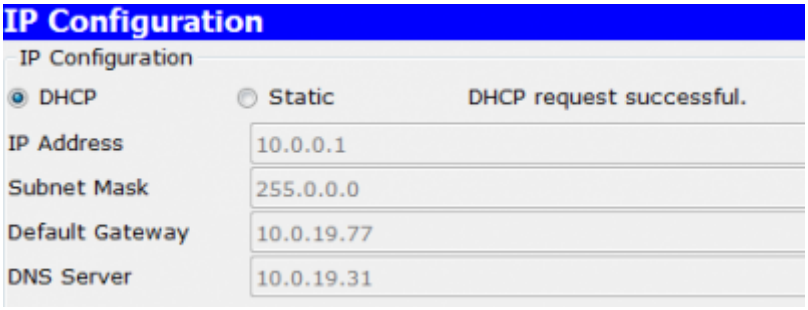
Dans cette partie nous allons nous prémunir contre la mise en réseau d'un serveur DHCP sauvage qui tenterait de se faire passer pour la passerelle par défaut dans la configuration de ses clients.

Situation initiale

Le PC du secrétariat (Secr1) récupère sa configuration réseau auprès du serveur DHCP comme indiqué sur ce schéma :



Voici sa configuration :



Notre collaborateur indélicat utilise son ordinateur personnel, change l'adresse MAC de l'interface pour la faire correspondre avec l'interface d'un PC habituellement connecté à la prise murale dans un bureau de la direction.

Il installe un serveur DHCP de type isc-dhcp-server avec les paramètres suivants :

- OS : Debian 8 modifiée
- DHCPD : isc-dhcp-server
- Adresse IP : 10.0.10.10
- Configuration pour les clients:
  - Intervalle d'adresses attribuées : 10.1.0.0 - 10.1.100.100 (/8)
  - Passerelle : 10.0.10.10 (lui-même)
  - domaine : corpora.te

Dans cette configuration, la situation précédente se reproduirait : attribution de cette configuration pour les clients et sniffing pour capturer tous les paquets sortant du réseau.

Or, l'administrateur réseau de Corporate utilise le DHCP snooping[1)] de Cisco sur les commutateurs de l'entreprise de telle sorte qu'il autorise la présence d'un seul serveur DHCP sur le réseau local.

Configurons le commutateur SW1 :

```
SW1(config)#ip dhcp snooping
SW1(config)#ip dhcp snooping vlan 1
SW1(config)#interface range fastEthernet 0/23-24
SW1(config-if-range)#ip dhcp snooping trust
SW1(config)#interface range fastEthernet 0/1-22
SW1(config-if-range)#no ip dhcp snooping trust
```

Nous avons autorisé la présence du service DHCP en provenance des ports fa0/23 et fa0/24 (en cas de changement dans la configuration STP, les liens entre commutateurs pourraient éventuellement changer) et explicitement refusé tout le reste.

Mais en se limitant à la configuration du commutateur SW1, on autorise implicitement notre pirate à faire fonctionner son serveur DHCP depuis les commutateurs SW2 et SW3

Il faut donc paramétrer tous les commutateurs du réseau. Concernant SW2, le seul port fa0/2 (sur lequel le serveur DHCP est connecté directement) doit être autorisé :

```
SW2(config)#ip dhcp snooping
SW2(config)#ip dhcp snooping vlan 1
SW2(config)#interface fastEthernet 0/2
SW2(config-if)#ip dhcp snooping trust
SW2(config)#interface fastEthernet 0/1
SW2(config-if)#no ip dhcp snooping trust
SW2(config)#interface range fastEthernet 0/3-24
SW2(config-if-range)#no ip dhcp snooping trust
```

et concernant SW3 :

```
SW3(config)#ip dhcp snooping
SW3(config)#ip dhcp snooping vlan 1
SW3(config)#interface fastEthernet 0/1
SW3(config-if)#ip dhcp snooping trust
SW3(config)#interface fastEthernet 0/24
SW3(config-if)#ip dhcp snooping trust
SW3(config)#interface range fastEthernet 0/2-23
SW3(config-if-range)#no ip dhcp snooping trust
```

Et en principe, tout doit fonctionner à merveille .... sauf que nous avons un message d'erreur :

```
SW2(config)#00:53:47: %DHCP_SNOOPING-5-DHCP_SNOOPING_NONZERO_GIADDR:
DHCP_SNOOPING drop message with non-zero giaddr or option82 value on
untrusted port, message type: DHCP DISCOVER, MAC sa: 000D.BD99.B487
```

En bloquant les ports non habilités à délivrer le service DHCP, nous avons bloqué non seulement les trames DHCP OFFER mais aussi DHCP DISCOVER des clients!

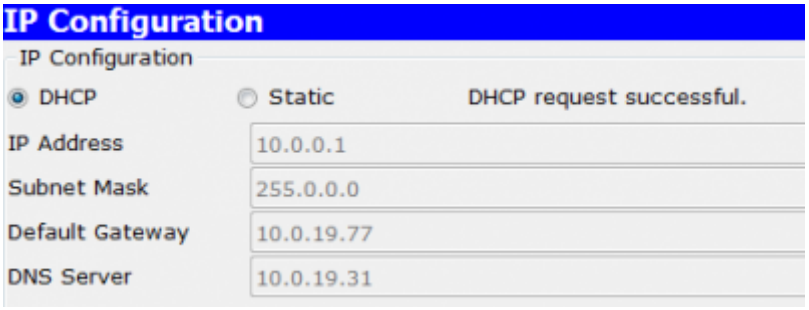
Il faut donc rajouter une option pour permettre l'attribution des configuration réseau **vers** les ports sur lesquels sont connectés les clients:

```
no ip dhcp snooping information option
```

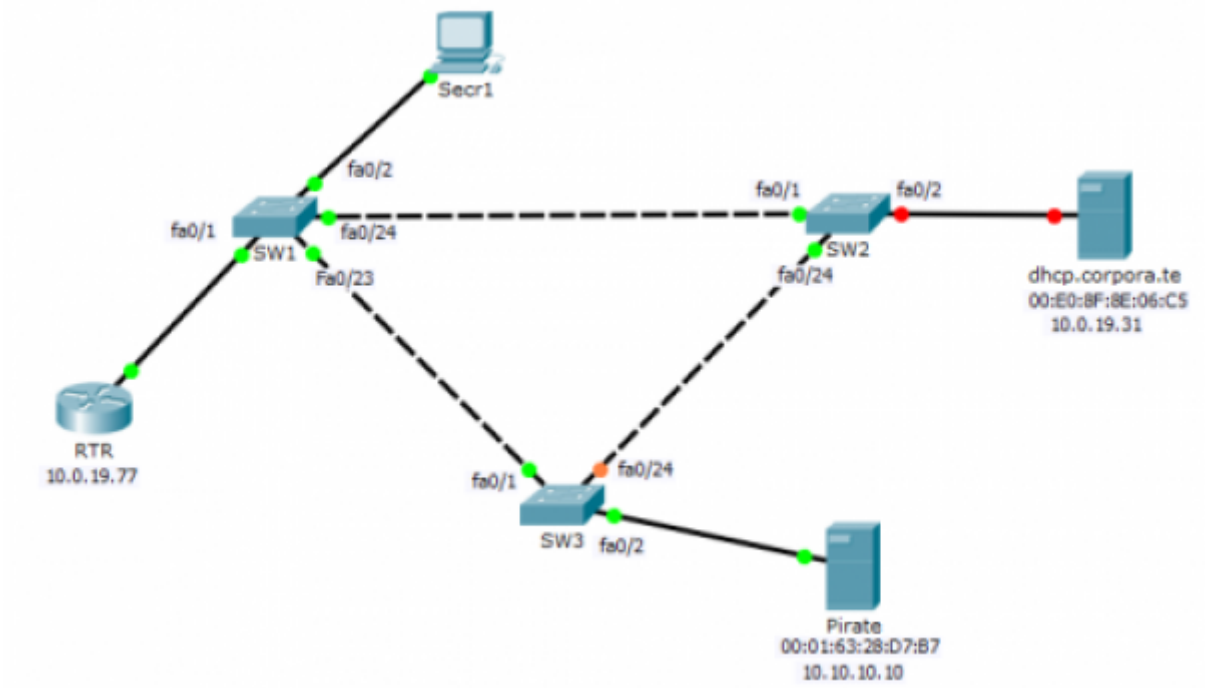
sur tous nos commutateurs.

Démonstration

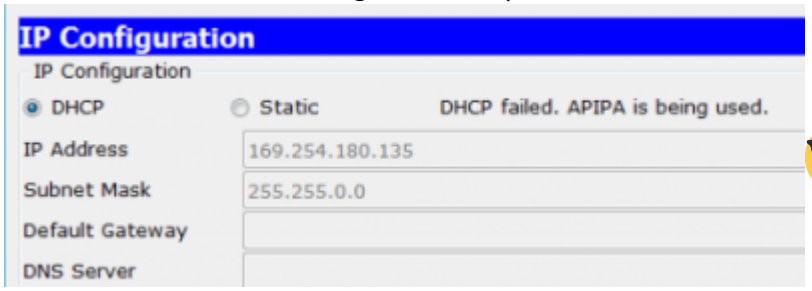
Nous redemandons une configuration depuis Secr1:



Notre pirate active son serveur DHCP et pour la forme on déconnecte le serveurDHCP autorisé (*dans un but de simplification de la démonstration*):



Et redemandons une configuration depuis Secr1:



😎 CQFD!

Le fichier Packet Tracer est téléchargeable sur ce lien pour de vérifier le bon fonctionnement du serveur DHCP de notre pirate sur le port fa0/2 de SW2 par exemple.

Surveillance

Sans la présence de port security restrictif, cela ne nous protège pas encore d'une attaque du type homme du milieu rencontrée dans le chapitre précédent. En effet, avec la bonne adresse MAC, l'interface connectée sur le port autorisé, on peut refaire les mêmes manipulations et aboutir aux mêmes résultats si la configuration de port security est trop laxiste (maximum d'adresses autorisées trop important).

C'est pourquoi, nous allons utiliser les solutions Cisco suivantes en complément de DHCP snooping et port security:

- IP source guard[2] *(on déborde légèrement sur la couche 3 mais dans le cadre de DHCP snooping nous ne sommes pas hors sujet)*;
- Dynamic ARP inspection[3].

IP source guard nous permet :

- D'attribuer manuellement une adresse MAC et une adresse IP d'hôte autorisé à un port.
- De vérifier que l'adresse IP d'une interface a bien été distribuée par le serveur DHCP (ainsi que la durée du bail).
- De vérifier la correspondance Adresse MAC/Adresse IP/Port en s'appuyant sur port security

DAI vérifie que les interfaces connectées aux ports du commutateur ont bien fait l'objet d'une configuration réseau par le serveur DHCP en contrôlant les adresses MAC des interfaces présentes dans la table de DHCP snooping.

On peut ainsi interdire l'usurpation d'adresses MAC à la volée à l'origine de l'attaque de l'homme du milieu. Ces deux solutions proposent également une option de log des alertes émises afin de surveiller l'activité du réseau.

Démonstration

1) IP source guard

Vérifions au préalable que DHCP snooping fonctionne correctement en regardant sa table d'attribution d'adresses:

```
SW1#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN
Interface
------------------  ---------------  ----------  -------------  ----  ------
--------------
F4:F2:6D:1E:C5:40   10.1.0.0         576         dhcp-snooping   1
FastEthernet0/2
Total number of bindings: 1
```

Activons maintenant IP source guard sur les interfaces des commutateurs à l'exception de elles qui relient les commutateurs entre eux:

```
SW1(config)#interface range fastEthernet 0/1-22
SW1(config-if-range)#ip verify source
```

et vérifions le résultat :

```
SW1#show ip verify source
```

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address
Vlan
---------  -----------  -----------  ---------------  -----------------  ---
-
Fa0/1      ip           inactive-no-snooping-vlan
Fa0/2      ip           active       10.1.0.0                            1
Fa0/3      ip           inactive-no-snooping-vlan
[...]
```

Si notre pirate en herbe s'attribue tout seul l'ip 10.10.10.10 en se connectant sur le port fa0/7, voilà le résultat sur le commutateur:

```
Mar  1 00:26:28.273: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on
Fa0/7, vlan
1.([000c.2986.eb1e/10.10.10.10/0000.0000.0000/10.0.19.31/00:26:28 UTC Mon
Mar 1 1993])
```

et le pirate ne peut pas communiquer avec le réseau.

Si port security est activé, on lui rajoute la commande suivante :

```
SW1(config)#interface range fastEthernet 0/1-22
SW1(config-if-range)#ip verify source port-security
```

ce qui nous permet de contrôler l'association IP/MAC/PORT :

```
SW1#sh ip verify source
Interface  Filter-type  Filter-mode  IP-address       Mac-address
Vlan
---------  -----------  -----------  ---------------  -----------------  ---
-
Fa0/1      ip           inactive-no-snooping-vlan
Fa0/2      ip-mac       active       10.1.0.0         F4:F2:6D:1E:C5:40  1
```

Tout ce qui “sort des clous” est jeté : voilà un bon début pour freiner notre pirate!

2) DAI

Partons du principe que notre pirate possède une adresse IP remise par le serveur DHCP, l'adresse MAC correspondante, bref il est en règle. Néanmoins cela ne va pas l'empêcher de jouer à l'homme du milieu avec Ettercap! Pour contrer ses ardeurs nous allons l'empêcher d'usurper les adresses MAC de ses voisins avec DAI.

le PC Secr1 est toujours présent ainsi que notre PC pirate qui a récupéré une adresse depuis le serveur DHCP:

```
SW1#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN
Interface
------------------  ---------------  ----------  -------------  ----  ------
--------------
F4:F2:6D:1E:C6:40   10.1.0.0         576         dhcp-snooping   1
FastEthernet0/3
00:0C:29:86:EB:1E   10.1.0.1         443         dhcp-snooping   1
FastEthernet0/7
Total number of bindings: 2
```

Nous activons DAI :

```
SW1(config)#ip arp inspection vlan 1
```

Nous indiquons à DAI que les ports du commutateur reliés à d'autres commutateurs ne sont pas concernés par son action:

```
SW1(config)#interface range fastEthernet 0/23-24
SW1(config-if-range)#ip arp inspection trust
SW1(config-if-range)#do show ip arp inspection interfaces
[...]
Fa0/22          Untrusted              15                1
Fa0/23          Trusted              None              N/A
Fa0/24          Trusted              None              N/A
[...]
```

et ... c'est tout!

On vérifie l'activation de DAI :

```
SW1#sh ip arp inspection vlan 1
[...]
 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
    1     Enabled          Active

 Vlan     ACL Logging      DHCP Logging      Probe Logging
 ----     -----------      ------------      -------------
    1     Deny             Deny              Off
```

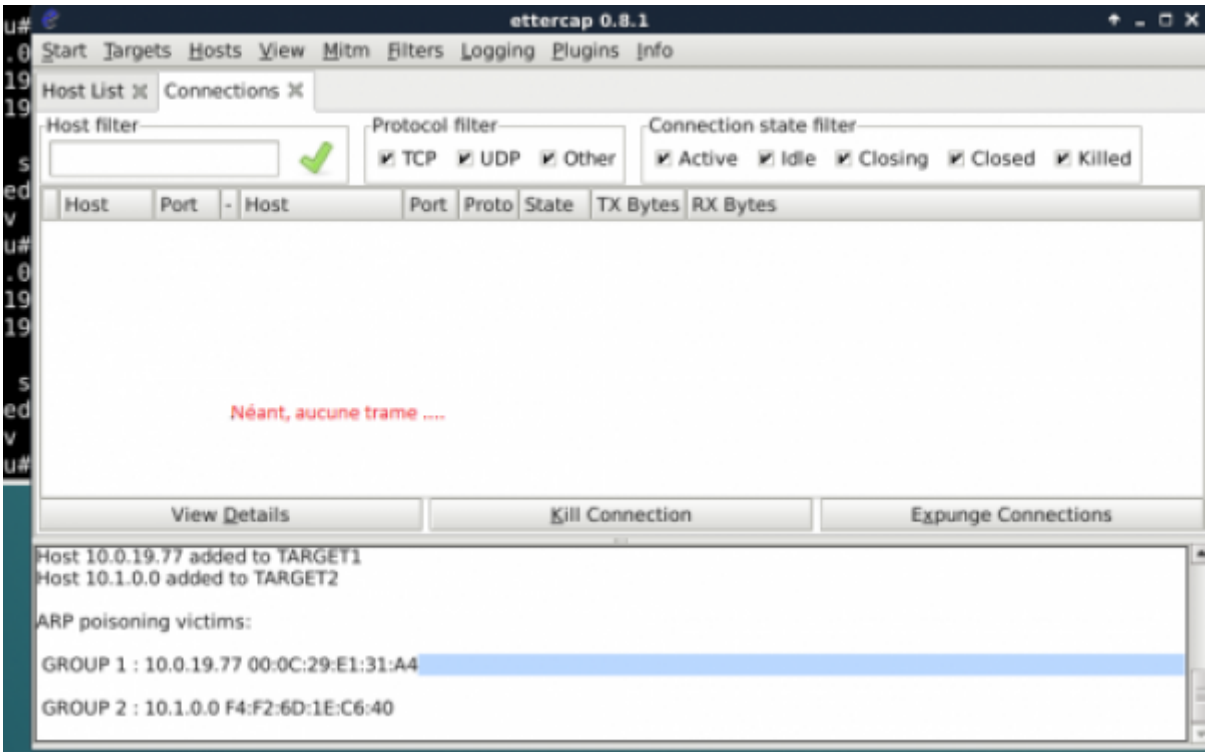
Toute interface dont l'adresse MAC ne correspond plus à l'information reçue préalablement par le serveur DHCP est rejetée.

On obtient les informations suivantes sur le commutateur lors d'une tentative d'attaque par la technique de l'homme du milieu :

```
Mar  1 00:33:57.332: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on
```

```
Fa0/7, vlan 1.  ([000c.2986.eb1e/10.1.0.1/0000.0000.0000/10.0.19.31/00:33:57
UTC Mon Mar 1 1993])
Mar  1 00:33:58.339: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on
Fa0/7, vlan 1. ([000c.2986.eb1e/10.1.0.1/0000.0000.0000/10.0.19.31/00:33:58
UTC Mon Mar 1 1993])
```

Notre pirate quant à lui, ne voit plus grand chose :



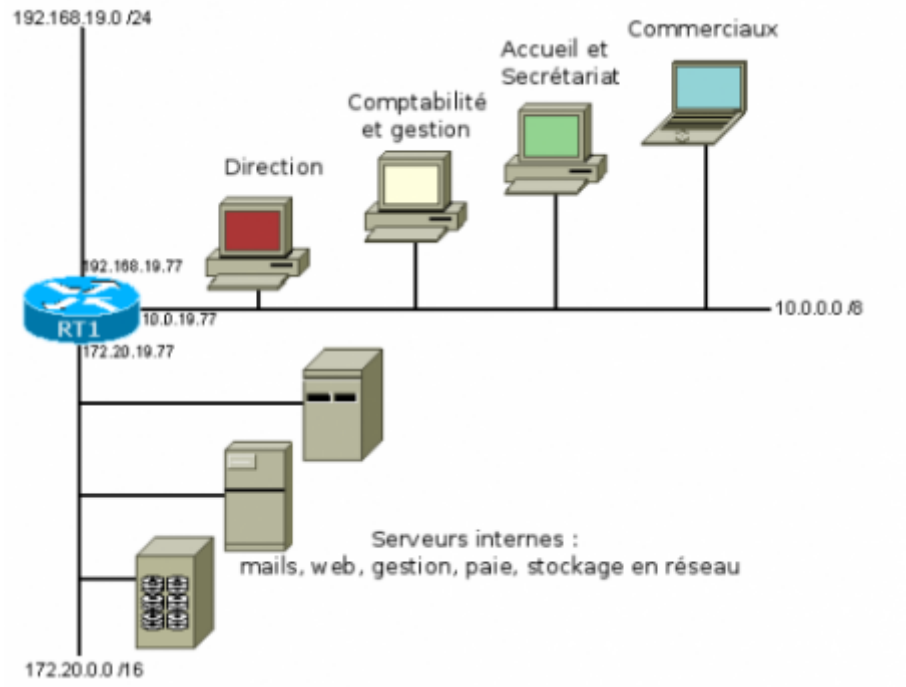
Conclusion

Avec un serveur DHCP, DAI et DHCP snooping activés, nous sommes en mesure d'empêcher les attaques de type homme du milieu et le fonctionnement de serveurs DHCP pirates. Couplés à IP source guard configuré automatiquement ou manuellement, nous sommes en mesure de contrôler l'ensemble des connexions au réseau. Néanmoins, cela demande un travail préparatoire conséquent et manque de souplesse, en particulier avec des postes de travail nomades susceptibles de devoir se connecter au réseau à la volée.

VLAN

Situation initiale

Notre domaine de diffusion est séparé des autres réseaux par le routeur RT1 :



Il peut arriver néanmoins qu'on ait besoin de scinder logiquement notre réseau en plusieurs réseaux distincts sans modifier physiquement l'emplacement des équipements.

Pour cela nous allons utiliser des VLANS[4)] définis par la norme IEEE 802.1q.

Les trames émises par un hôte situé dans un VLAN ne seront reçues que par les hôtes situés dans le même VLAN, même si ils sont physiquement connectés au même commutateur.

Concernant la couche 2, nous verrons 2 sortes de VLAN :

- VLAN par port
- VLAN par adresse MAC

(il existe aussi des VLAN de niveau 3, ce sont des VLAN d'adresses réseaux que nous n'aborderons pas dans le cadre de ce wiki).

VLAN par port ("de niveau 1")

Pour créer des VLAN par port (dit “de niveau 1”), il suffit d'affecter à chaque port concerné un numéro de VLAN.

Avantages :

- Extrême simplicité
- Déploiement des VLAN sur plusieurs commutateurs

Inconvénients :

- Plutôt limité en terme de sécurité : il suffit de se connecter sur une prise réseau reliée au port affecté au VLAN qu'on souhaite rejoindre!
- Si on doit changer le branchement d'un ordinateur, on doit également reconfigurer le port concerné.

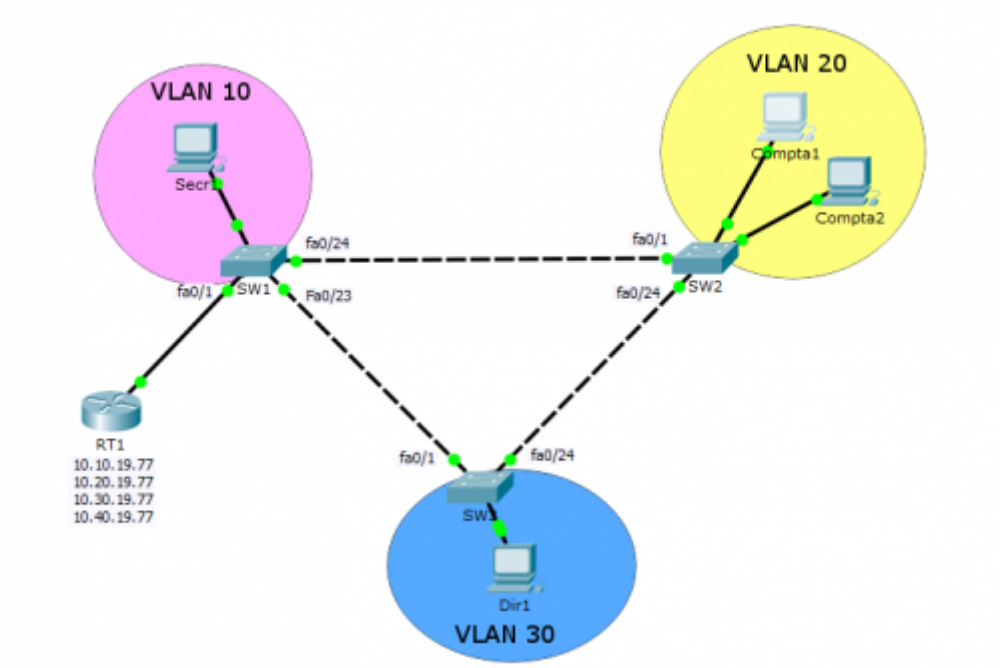
Démonstration

Nous souhaitons arriver à la situation suivante, où chaque service se trouve dans un VLAN différent :

```
* Secrétariat : VLAN 10, adresses de 10.10.0.0 à 10.10.255.255
* Comptabilité : VLAN 20, adresses de 10.20.0.0 à 10.20.255.255
* Direction : VLAN 30, adresses de 10.30.0.0 à 10.30.255.255
* Commerciaux : VLAN 40, adresses de 10.40.0.0 à 10.40.255.255
```

Ayant un nombre limité de ports sur SW1, nous utiliserons l'étiquetage des trames transitant par le port fa0/1 du commutateur (mode trunk) avec leur numéro de VLAN pour pouvoir les différencier sur une connexion commune vers le routeur RT1 (lui même capable d'encapsuler les trames taguées). Les ports des commutateurs reliés aux autres commutateurs seront également tagués en mode trunk.

Le routeur RT1 nous servira de serveur DHCP pour chaque VLAN dans la mesure où il intègre l'encapsulation 802.1q. Voici la configuration souhaitée :



Nous configurons le commutateur SW1 :

```
SW1(config)#vlan 10
SW1(config-vlan)#name secretariat
SW1(config)#exit
SW1(config)#interface fa0/1
SW1(config-if)#switchport mode trunk
SW1(config-if)#interface range fa0/23-24
SW1(config-if-range)#switchport mode trunk
SW1(config-if-range)interface range fa0/2-22
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
```

Voilà notre VLAN “secretariat” créé et les ports correctement configurés. Etant feignant par nature, je

vais déclarer tous mes VLAN au commutateur SW1 et ensuite propager ces informations à tous les autres commutateurs grâce au protcole VTP[5]:

```
SW1(config)#vlan 20
SW1(config-vlan)#name comptabilite
SW1(config)#exit
SW1(config)#vlan 30
SW1(config-vlan)#name direction
SW1(config)#exit
SW1(config)#vlan 40
SW1(config-vlan)#name commerciaux
SW1(config)#exit
SW1(config)#vtp domain corpora.te
Changing VTP domain name from NULL to corpora.te
SW1(config)#vtp password plop
Setting device VLAN database password to plop
SW1(config)#vtp mode server
Setting device to VTP SERVER mode.
```

Nous sommes prêts à transmettre nos VLAN au reste des commutateurs qu'il va falloir configurer pour recevoir ces informations. SW2 :

```
SW2(config)#int fa0/1
SW2(config-if)#switchport mode trunk
SW2(config)#int fa0/24
SW2(config-if)#switchport mode trunk
SW2(config)#vtp domain corpora.te
Changing VTP domain name from NULL to corpora.te
SW2(config)#vtp password plop
Setting device VLAN database password to plop
SW2(config)#vtp mode client
Setting device to VTP client mode.
```

Vérifions la bonne prise en compte des VLAN :

```
  VLAN Name                              Status    Ports
---- -------------------------------- --------- -------------------------------
---
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12,
Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16,
Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20,
Fa0/21
                                                Fa0/22, Fa0/23, Gig0/1,
Gig0/2
10   secretariat                      active
20   comptabilite                     active
30   direction                        active
```

```
40   commerciaux                      active
```

Nous pouvons attribuer les interfaces restantes au VLAN 20 :

```
SW2(config)#interface range fa0/2-23
SW2(config-if-range)#switchport mode access
SW2(config-if-range)#switchport access vlan 20
```

Il nous reste plus qu'à configurer le commutateur SW3 :

```
SW3(config)#int fa0/1
SW3(config-if)#switchport mode trunk
SW3(config)#int fa0/24
SW3(config-if)#switchport mode trunk
SW3(config)#vtp domain corpora.te
Changing VTP domain name from NULL to corpora.te
SW3(config)#vtp password plop
Setting device VLAN database password to plop
SW3(config)#vtp mode client
Setting device to VTP client mode.
SW3(config)#interface range fa0/2-23
SW3(config-if-range)#switchport mode access
SW3(config-if-range)#switchport VLAN 30
```

Les VLAN étant tous configurés sur les commutateurs, il faut maintenant confgurer le routeur RT1 pour délivrer les configurations réseaux au hôtes situés sur chaque VLAN.

Pour cela nous allons utiliser la technique d'encapsulation 802.1q pour éviter d'avoir 4 connexions physiques occupées par le routeur:

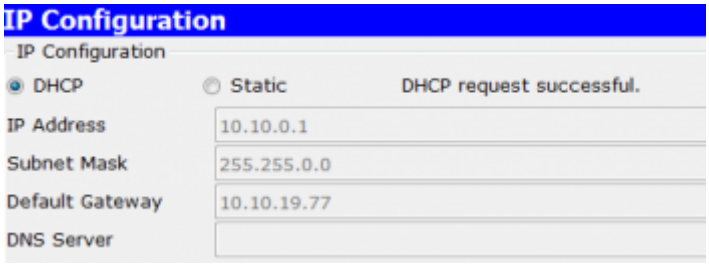
```
RT1(config)#interface fastEthernet0/0.10
RT1(config-subif)#encapsulation dot1Q 10
RT1(config-subif)#ip address 10.10.19.77 255.255.0.0
RT1(config-subif)#interface fastEthernet0/0.20
RT1(config-subif)#encapsulation dot1Q 20
RT1(config-subif)#ip address 10.20.19.77 255.255.0.0
RT1(config-subif)#interface fastEthernet0/0.30
RT1(config-subif)#encapsulation dot1Q 30
RT1(config-subif)#ip address 10.30.19.77 255.255.0.0
RT1(config-subif)#interface fastEthernet0/0.40
RT1(config-subif)#encapsulation dot1Q 40
RT1(config-subif)#ip address 10.40.19.77 255.255.0.0
RT1(config-subif)#interface fastEthernet0/0
RT1(config-subif)#no sh
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
%LINK-5-CHANGED: Interface FastEthernet0/0.10, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.10, changed
state to up
%LINK-5-CHANGED: Interface FastEthernet0/0.20, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.20, changed
state to up
%LINK-5-CHANGED: Interface FastEthernet0/0.30, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.30, changed
state to up
%LINK-5-CHANGED: Interface FastEthernet0/0.40, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.40, changed
state to up
```

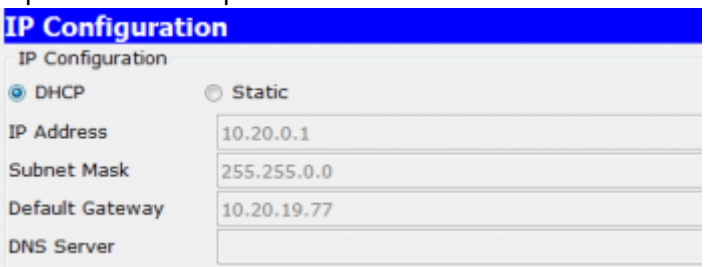
Voilà un routeur configuré! Il ne reste plus qu'à mettre en place le serveur DHCP :

```
RT1(config)#ip dhcp pool secretariat
RT1(dhcp-config)#network 10.10.0.0 255.255.0.0
RT1(dhcp-config)#default-router 10.10.19.77
RT1(dhcp-config)#network 10.20.0.0 255.255.0.0
RT1(dhcp-config)#default-router 10.20.19.77
RT1(dhcp-config)#ip dhcp pool direction
RT1(dhcp-config)#network 10.30.0.0 255.255.0.0
RT1(dhcp-config)#default-router 10.30.19.77
RT1(dhcp-config)#ip dhcp pool commerciaux
RT1(dhcp-config)#network 10.40.0.0 255.255.0.0
RT1(dhcp-config)#default-router 10.40.19.77
RT1(dhcp-config)#exit
RT1(config)#ip dhcp excluded-address 10.10.19.77 10.20.19.77
RT1(config)#ip dhcp excluded-address 10.30.19.77 10.40.19.77
```

Notre serveur DHCP est paramétré, nous allons demander une configuration réseau du client Secr1 dans le VLAN 10 :



Testons la même chose depuis le PC Compta1 sur le VLAN 20 :



En revanche maintenant que nous routons entre tous les VLAN, les flux ne sont plus isolés, comme le montre un ping depuis compta1 vers secr1 :

```
Pinging 10.10.0.1 with 32 bytes of data:
Reply from 10.10.0.1: bytes=32 time=11ms TTL=127
Reply from 10.10.0.1: bytes=32 time=0ms TTL=127
```

Il faut donc mettre quelques droits sur le routeur pour séparer les vlans :

```
RT1(config)#ip access-list extended vlan
RT1(config-ext-nacl)#permit ip 10.10.0.0 0.0.255.255 10.10.0.0 0.0.255.255
RT1(config-ext-nacl)#permit ip 10.20.0.0 0.0.255.255 10.20.0.0 0.0.255.255
RT1(config-ext-nacl)#permit ip 10.30.0.0 0.0.255.255 10.30.0.0 0.0.255.255
RT1(config-ext-nacl)#permit ip 10.40.0.0 0.0.255.255 10.40.0.0 0.0.255.255
RT1(config)#interface fa0/0.10
RT1(config-subif)#ip access-group vlan in
RT1(config-subif)#interface fa0/0.20
RT1(config-subif)#ip access-group vlan in
RT1(config-subif)#interface fa0/0.30
RT1(config-subif)#ip access-group vlan in
RT1(config-subif)#interface fa0/0.40
RT1(config-subif)#ip access-group vlan in
```

Je l'ai avoué, je suis feignant: Cette méthode de “factorisation” du groupe d'ACL étendues affecté en commun à toutes les interfaces me plaît très moyennement après réflexion : si on rajoute une interface, un ou plusieurs VLAN, c'est moche …

Je propose donc une deuxième méthode ci-dessous, que je trouve peut être plus propre et plus souple pour la suite :

```
RT1(config)#access-list 1 permit 10.10.0.0 0.0.255.255
RT1(config)#access-list 2 permit 10.20.0.0 0.0.255.255
RT1(config)#access-list 3 permit 10.30.0.0 0.0.255.255
RT1(config)#access-list 4 permit 10.40.0.0 0.0.255.255
RT1(config)#interface fa0/0.10
RT1(config-subif)#ip access-group 1 out
RT1(config-subif)#interface fa0/0.20
RT1(config-subif)#ip access-group 2 out
RT1(config-subif)#interface fa0/0.30
RT1(config-subif)#ip access-group 3 out
RT1(config-subif)#interface fa0/0.40
RT1(config-subif)#ip access-group 4 out
```

Nous avons autorisé les trames d'un VLAN à lui-même et vers les interfaces du routeur (access group out). Verification :

```
PC>ping 10.10.0.1
Pinging 10.10.0.1 with 32 bytes of data:
Reply from 10.20.19.77: Destination host unreachable.
Reply from 10.20.19.77: Destination host unreachable.
Reply from 10.20.19.77: Destination host unreachable.
Reply from 10.20.19.77: Destination host unreachable.
```

Conclusion

Parfaits pour scinder logiquement un domaine de diffusion, les VLAN par port permettent par exemple d'interdire l'accès au réseau de la direction depuis les bureaux du secrétariat.

Néanmoins cela perd toute efficacité si notre pirate branche son ordinateur sur une prise murale dans un des bureaux de la direction ... Qui plus est, on doit reparamétrer les commutateurs en cas de changement de bureau d'un collaborateur d'un autre service, ce qui à terme peut vite devenir pénible avec des ordinateurs portables nomades.

Ici se trouve le fichier Packet Tracer avec la configuration (VLAN) décrite dans ce chapitre

VLAN par adresse MAC ("de niveau 2")

Nous reprendrons la même configuration que dans la paragraphe précédent:

- VLAN 10 : secretariat
- VLAN 20 : comptabilite
- VLAN 30: direction
- VLAN 40 : commerciaux

La différence est que nous n'allons **pas scinder** le domaine de diffusion **par ports** mais grâce à l'**adresse MAC** des interfaces qui se connectent aux commutateurs (ces VLAN sont appelés “VLAN de niveau 2”). Il faut donc posséder des équipements de niveau 2 qui le permettent, ce qui pas systématique.

L'intérêt est évident : on peut déplacer à volonté un ordinateur nomade, peut importe où son interface réseau est branchée, l'hôte sera dans le bon VLAN ! L'inconvénient est tout aussi évident : il faut maintenir à jour une table des adresses MAC, ce qui risque d'être fastidieux avec de nombreux hôtes.
Avec nos commutateurs Cisco 2960, nous pouvons créer des VLAN de niveau 2 en nous appuyant sur VMPS[6]. Le principe est simple : le commutateur s'adresse à un serveur qui détient une base d'adresses MAC autorisées et le VLAN correspondant à chaque interface.

Côté commutateur, la configuration des ports est simple:

```
SW1(config)# int range fa0/2-22
SW1(config-if-range)#switchport mode dynamic
```

et nous paramétrons ensuite les coordonnées du serveur VMPS :

```
SW1(config)# vmps server 172.16.29.29 primary
SW1(config)# vmps server 172.16.29.30
SW1(config)# vmps retry 5
SW1(config)# vmps reconfirm 60
```

Nous avons indiqué un serveur primaire, un secondaire au cas où le commutateur n'arriverait pas à communiquer avec le serveur primaire au bout de 5 tentatives, et le délai (1 heure) servant en quelque sorte de bail entre deux requêtes au serveur.

Côté serveur, nous pouvons utiliser OpenVMPS [7] sur linux debian, le fichier de configuration assez trivial dans le cas présent se trouve dans /etc/vmpsd.conf:

```
!--- initialisation
vmps mode open
vmps domain corpora.te
```

```
vmps no-domain-req deny
vmps-mac-addrs
!--- declaration des VLAN:
!VLAN secretariat
address  F4F2.6D1E.C640 vlan-name secretariat
!VLAN comptabilite
address 000C.2986.EB1E vlan-name comptabilite
address 000C.2986.AB02 vlan-name comptabilite
!VLAN direction
address 000C.2877.14E1 vlan-name direction
!--- fin de la configuration
```

Mais cette solution est faillible : il suffit à notre pirate d'usurper l'adresse MAC d'un PC situé dans le VLAN convoité pour y être connecté.

Conclusions

Vous êtes enfin arrivé(e) au bas de la page!!

Après avoir vu l'intérêt du port security, du DHCP snooping couplé a Dynamic ARP Inspection et IP Source Guard, les VLAN par port et par adresse MAC, nous avons pu voir les avantages et les inconvénients de chaque solution.

Une problématique commune se dégage : Comment assurer **efficacement** la sécurité de mon réseau sans **y passer tout mon temps** ??

Une réponse se trouve dans le chapitre suivant!

1)
DHCP Snooping
2)
Cisco IP source Guard
3)
Dynamic ARP inspection
4)
Virtual Local Area Network
5)
VLAN Trunking Protocol
6)
VLAN Management Policy Server
7)
OpenVMPS sur SourceForge