	<p align="center">Bloc 3 - Chap 3 ARP poisoning et Man In The Middle</p>	<p align="center">E. Le Gars</p>
<p align="center">B3-S2</p>	<p align="center">Eléments de Cours + Travaux Pratiques sous Labtainer</p>	<p align="center">BTS1-S2</p>

Objectifs :

- Rappels sur les mécanismes ARP
- Introduire les principes d'une attaque par ARP poisoning / spoofing
- Comprendre les principe d'un Man in The Middle (MITM)
- Mettre en évidence les vulnérabilités d'une infrastructure aux attaques de type MITM
- Méthodes de limitation et de prévention des attaques par ARP poisoning

Au Menu

Présentation de l'environnement (sous Labtainer)	2
2. Principe de la pratique du "Man in the Middle"	3
3. Déroulement de la Manip	5
3.1 Préparation de l'attaquant	5
3.2 Avant l'attaque : Monitorer le trafic transitant via l'attaquant	5
3.3 Mener l'attaque d'ARP spoofing sur l'utilisateur et la gateway	6
4. Contre mesures face aux attaques par ARP poisoning/spoofing	8
4.1 Segmenter le réseau en VLANs	8
4.2 Sur les commutateurs : Fonction DHCP Snooping et Deep ARP inspection	8
4.3 Sur les hôtes : Mettre en place une table ARP statique	9

Consignes :

Les éléments **marqués en vert** sont des commandes à rentrer sur les systèmes de la manip

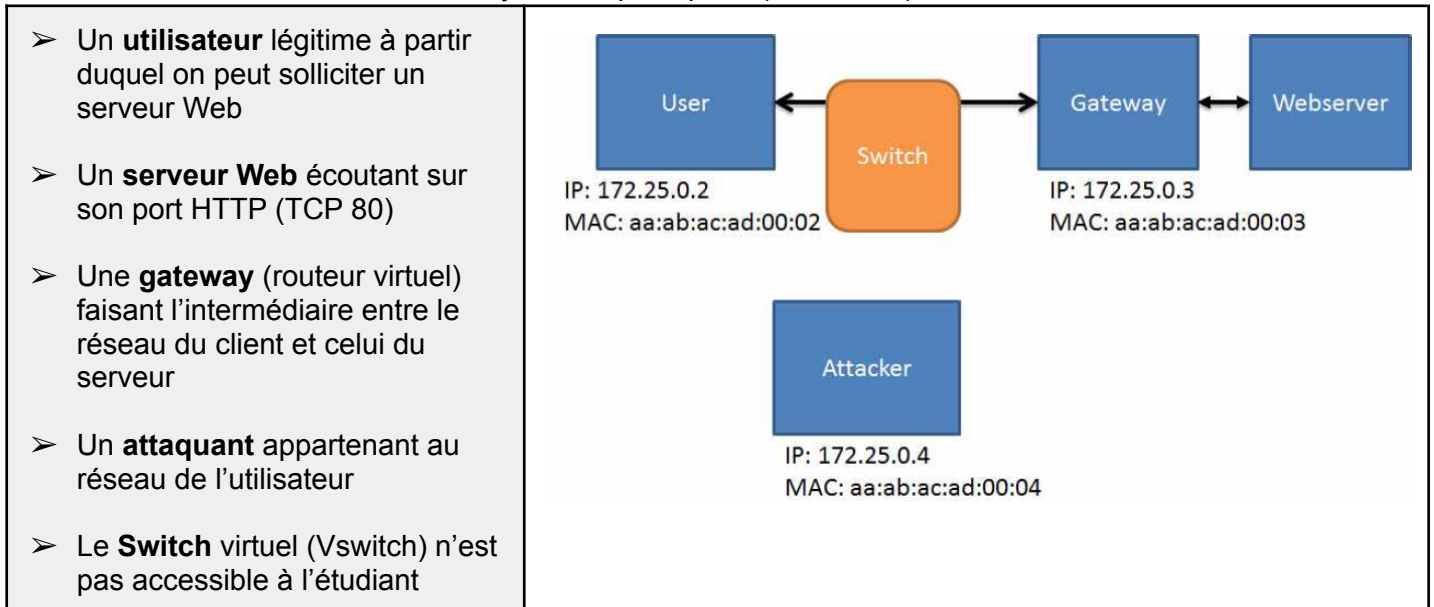
Les éléments **marqués en bleu** sont relatifs à des questions auxquelles vous devez répondre dans les encadrés prévus à cet effet

Une **attaque de l'homme du milieu** (MITM, Man In The Middle) dans un réseau local désigne un procédé permettant d'insérer artificiellement un hôte "espion" sur le passage de **trames** ayant normalement lieu entre 2 hôtes légitimes (ou entre un hôte légitime et sa passerelle).

Dans ce type d'attaque, l'attaquant envoie de faux paquets ARP aux 2 hôtes légitimes afin de se faire passer pour l'un auprès de l'autre, et inversement. L'attaquant peut ainsi intercepter ou manipuler leur trafic de données. Cette attaque est nécessaire pour procéder à une autre attaque déjà vue précédemment : Le DNS poisoning

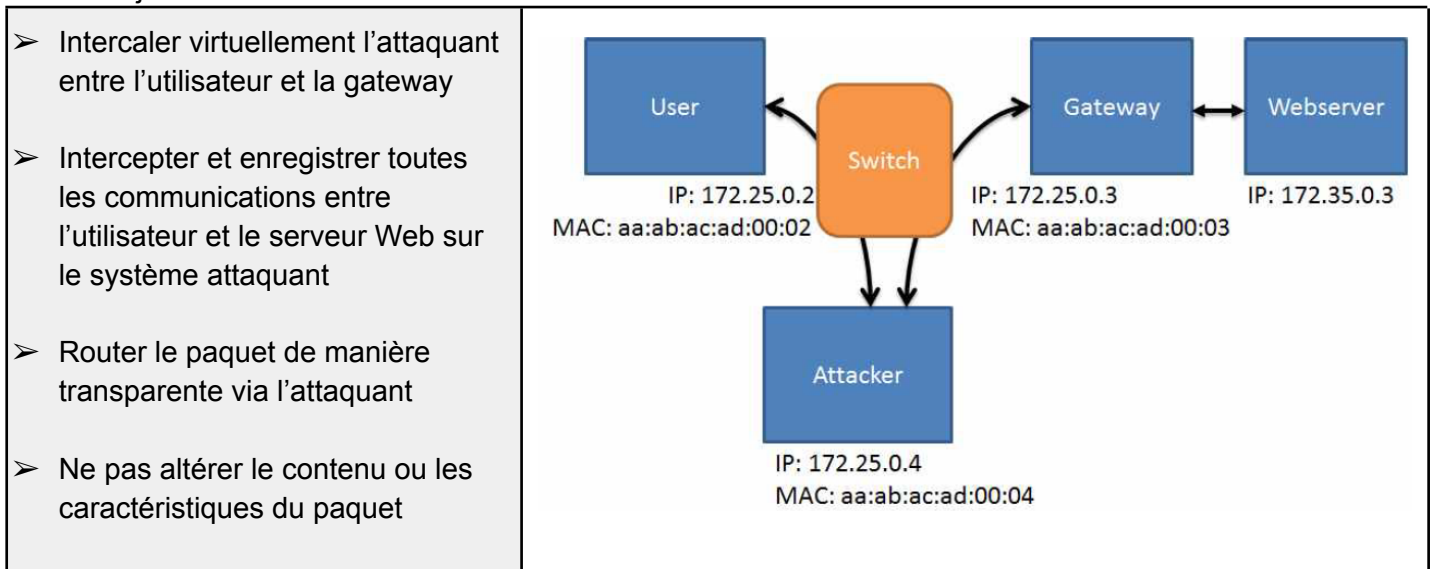
1. Présentation de l'environnement (sous Labtainer)

- L'environnement sera constitué de 4 systèmes principaux (containers) :



- On peut intervenir sur chacun de ces systèmes par l'intermédiaire de leur interface en ligne de commande

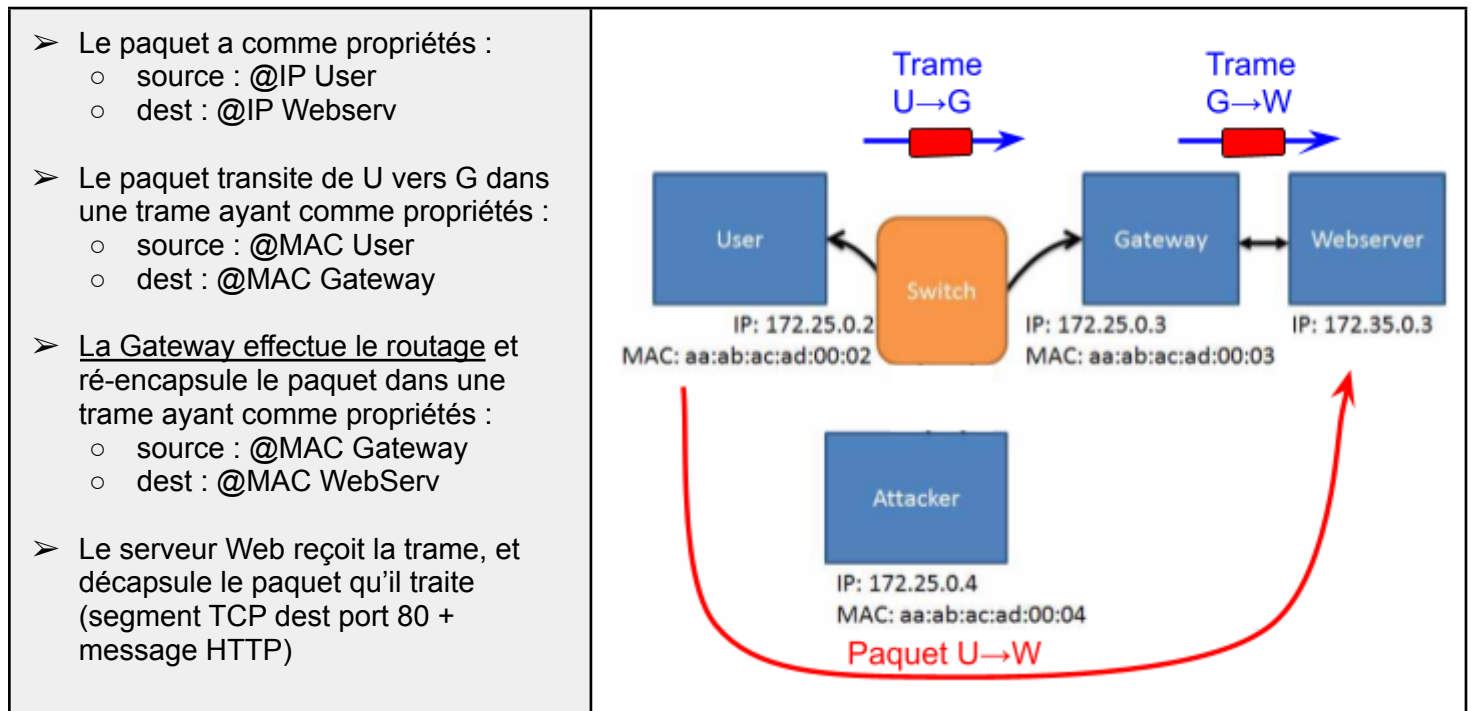
- Le but du jeu :



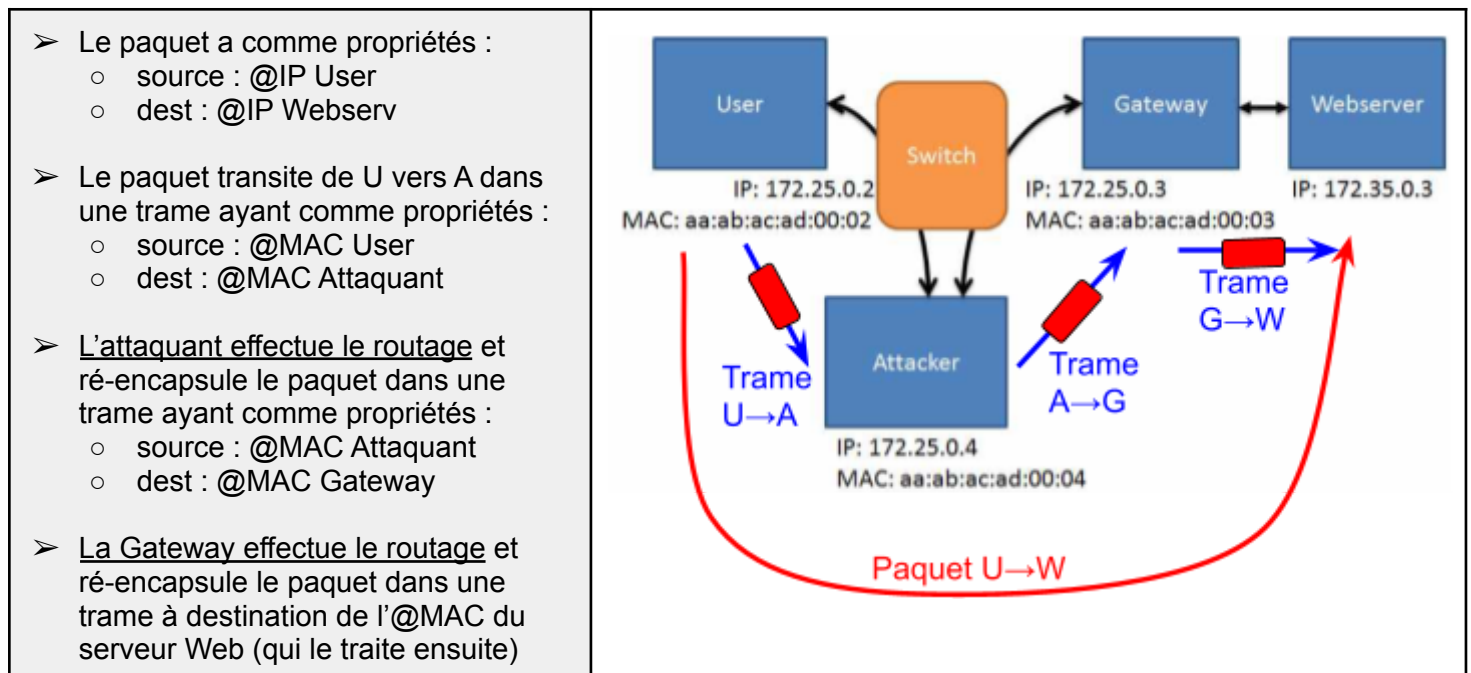
2. Principe de la pratique du “Man in the Middle”

- Nous allons effectuer une attaque de type “Man in The Middle” (MITM) en empoisonnant le cache ARP des deux systèmes communiquant par trames ethernet au sein du même réseau (systèmes User et Gateway)

En temps normal, le transit d'un paquet de l'utilisateur vers le Serveur Web est le suivant :



Lors d'une attaque par ARP spoofing, le transit d'un paquet de l'utilisateur vers le Serveur Web est le suivant :



- **Attention** : Ce procédé n'a rien à voir avec les attaques de type **MAC flooding** ou de **MAC spoofing** qui visent les **commutateurs**. Dans ces deux dernières, c'est la table MAC-Port (ou CAM table) d'un switch qui est visée :
 - Avec le **MAC flooding**, un attaquant sature la table MAC/Port d'un switch afin de l'empêcher de découvrir toute nouvelle correspondance MAC/Port → On peut immobiliser un LAN à court terme
 - Avec le **MAC spoofing**, un attaquant usurpe l'@MAC d'un hôte légitime et écrase l'enregistrement MAC/Port d'un switch concernant cet hôte → Les trames à destination de l'hôte légitime sont redirigées par le switch uniquement vers l'attaquant (et la victime ne reçoit rien de ce switch)

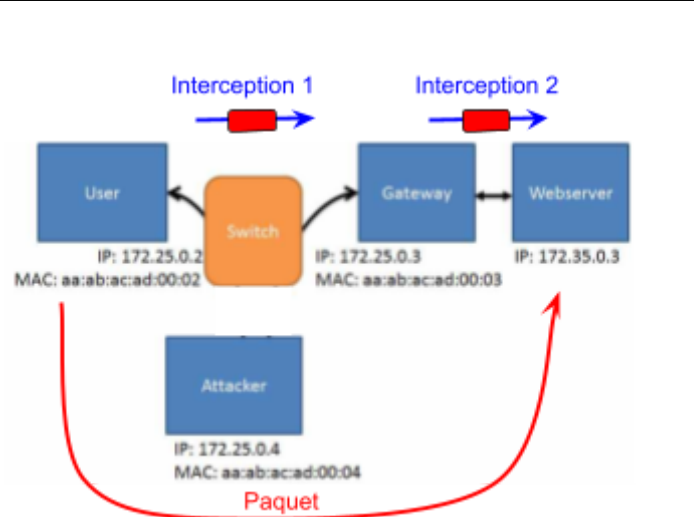
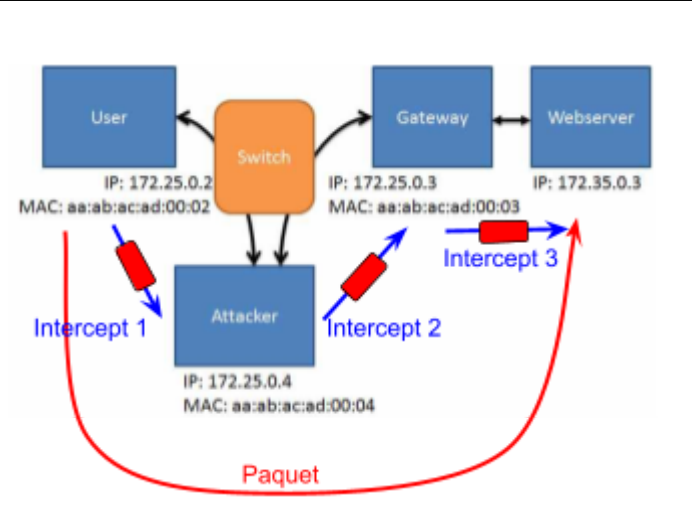
- Afin de “forcer” le trafic transitant normalement entre la gateway et l'utilisateur à passer au travers du système de l'attaquant :
 - L'attaquant doit se faire passer pour la gateway auprès de l'utilisateur
 - L'attaquant doit se faire passer pour l'utilisateur auprès de la gateway
- Pour ce faire, l'attaquant doit empoisonner les tables ARP de l'utilisateur et de la gateway, de manière à :
 - Faire croire à l'utilisateur qu'il envoie une trame à destination de la gateway alors qu'il s'agit en fait de l'attaquant
 - Faire croire à la gateway qu'elle envoie une trame à destination de l'utilisateur alors qu'il s'agit en fait de l'attaquant

➤ Etats des tables/cache ARP des systèmes avant et pendant l'attaque :

	Cache ARP Utilisateur	Cache ARP Gateway
Avant l'attaque	@IP Gateway ↔ @MAC Gateway	@IP Utilisateur ↔ @MAC Utilisateur
Pendant l'attaque	@IP Gateway ↔ @MAC Attaquant	@IP Utilisateur ↔ @MAC Attaquant

A noter : A aucun moment les caractéristiques du paquet ne changent : il a toujours comme source originelle l'utilisateur et comme destination finale le serveur Web !

Pour les 2 situations suivantes, renseigner les @MAC des trames et les @IP des paquets à tous les points d'interception lors d'un envoi de paquet à partir de l'utilisateur vers le Serveur Web (utiliser uniquement les noms des machines - pas de place pour les adresses)

<u>Avant</u> l'attaque ARP Spoofing			<u>Pendant</u> l'attaque ARP Spoofing		
					
	@Mac source → Dest	@IP source → Dest		@Mac source → Dest	@IP source → Dest
1	→	→	1	→	→
2	→	→	2	→	→
			3	→	→

3. Déroulement de la Manip

- Lancer la VM **LabtainerVM2** à partir de VirtualBox
- Configurer le mapping clavier en français (à faire à chaque redémarrage de la VM) : **setxkbmap fr**
- Lancer le labo **labtainer arp-spoof** depuis la session “student”
- 4 systèmes “conteneurs” sont lancés, préconfigurés et liés entre eux par un réseau virtuel
- Inutile de renseigner l’@Email (prévue uniquement pour publier les résultats d’une manip et la faire évaluer par un instructeur)

- Outils à utiliser sur le système attaquant :
 - Wireshark permettant d’analyser les trames transitant par le système attaquant
 - Outil **arpspoof** permettant d’usurper les communications de couches 2 et 3 (respectivement Ethernet et IPV4)

Question préliminaire : Rappeler la raison d’être et le principe des échanges ARP (requête + réponse)

3.1 Préparation de l’attaquant

- L’attaquant doit prendre le rôle d’un **routeur** et doit être paramétré en conséquence. On doit donc positionner le paramètre d’environnement “forwarding” à la valeur “1” avec la commande :
Sur l’attaquant : sysctl -w net.ipv4.conf.all.forwarding=1
- Vérifier que le paramètre “forwarding” est bien positionné : la commande **sysctl net.ipv4.conf.all.forwarding** doit retourner la valeur “1”
- L’outil **arpspoof** installé sur l’attaquant va permettre d’envoyer des **réponses ARP non sollicitées (ou gratuites)**, c’est-à-dire des correspondances @IP/@MAC pour lesquelles il n’aura pas explicitement fait de demandes.

Pourquoi l’attaquant n’attend-il pas simplement des requêtes ARP des ces 2 systèmes pour y répondre ?

A l’inverse, quelle est la faiblesse du procédé d’envoi de réponses ARP non-sollicitées ?

3.2 Avant l’attaque : Monitorer le trafic transitant via l’attaquant

- Avant de lancer l’ARP Spoofing, nous allons vérifier le trafic transitant par l’attaquant en lançant l’analyseur de paquets WireShark :
Sur l’attaquant : wireshark -ki eth0
- Depuis l’utilisateur, nous allons solliciter le serveur Web en effectuant une requête HTTP. Le système de l’utilisateur n’étant pas muni d’un client Web, nous allons utiliser la commande **wget** :
Sur l’utilisateur : wget <adresse du serveur Web>
- Le fichier **index.html** doit normalement être récupéré par l’utilisateur. En vérifier le contenu avec la commande
Sur l’utilisateur : more index.html

Quel Trafic l’attaquant a-t-il intercepté ? Pourquoi ?

Wireshark est-il à proprement parler un outil de type “sniffer” ?

3.3 Mener l'attaque d'ARP spoofing sur l'utilisateur et la gateway

➤ Nous allons maintenant procéder à l'ARP poisoning à proprement parler, à savoir :

- Envoyer des réponses ARP non-solicit  es (ou gratuites) au client en indiquant que l'@MAC de l'attaquant est celle de la passerelle →   crasement du *record* ARP de l'@IP de la passerelle sur le client
- Envoyer des r  ponses ARP non-solicit  es    la gateway en indiquant que l'@MAC de l'attaquant est celle de l'utilisateur →   crasement du *record* ARP de l'@IP du client sur la passerelle

➤ Les commandes    renseigner sont les suivantes :

Sur l'attaquant :

- **sudo arpspoof -t <User IP> <gateway IP>**
- **sudo arpspoof -t <gateway IP> <User IP>**

Attention, chaque commande doit   tre lanc  e dans une session distincte de l'attaquant, car l'ex  cution d'un processus peut bloquer la saisie d'une autre commande au sein de la m  me session.
→ La fen  tre de CLI comporte 3 onglets : un pour wireshark, une pour la 1  re commande, et une pour la 2  me commande) !

Alternativement / Pour aller plus loin : Pour ceux qui le d  sirent, on peut parfaitement ex  cuter plusieurs programmes en parall  le sur une unique session en la suivant d'un '&' ('et' commercial). Cela aura pour effet de lancer le programme en t  che de fond sans bloquer l'invite de la session. On peut lister les processus ainsi que leur identifiant (PID) avec une commande de type **ps -ef | grep <nom du processus>**. Pour arr  ter ces processus, vous lancerez la commande **kill <num  ro du processus ou PID>**

➤ Renouveler la requ  te HTTP depuis le client l  gitime :
Sur l'utilisateur : wget <adresse du serveur Web>

R  colter les trames transitant sur l'attaquant via Wireshark et prendre un snapshot des   changes de paquets entre le client l  gitime et le serveur Web :

R  cup  rer la table ARP du client l  gitime et de la passerelle pendant l'attaque, les comparer avec celle obtenues pr  c  demment et conclure :

	Cache ARP <u>avant</u> l'attaque	Cache ARP <u>pendant</u> l'attaque
User	@IP Gateway ↔ @MAC Gateway	@IP Gateway ↔ @MAC Attaquant
Gateway	@IP Gateway ↔ @MAC Gateway	@IP Gateway ↔ @MAC Gateway

Conclusion :

Pour quelle raison l'attaquant envoie une ARP "unsolicited" toutes les secondes, et non pas juste une seule fois ?

D'après vous, comment le switch peut-il s'apercevoir qu'il y a une supercherie ?

- Après avoir quitté Wireshark, vous pouvez maintenant fermer le labo avec la commande :
Sur la session "student" : stoplab arp-spoof

4. Contre mesures face aux attaques par ARP poisoning/spoofing

4.1 Segmenter le réseau en VLANs

Dans le cas d'une attaque de type MITM, toutes les machines doivent appartenir au même LAN :

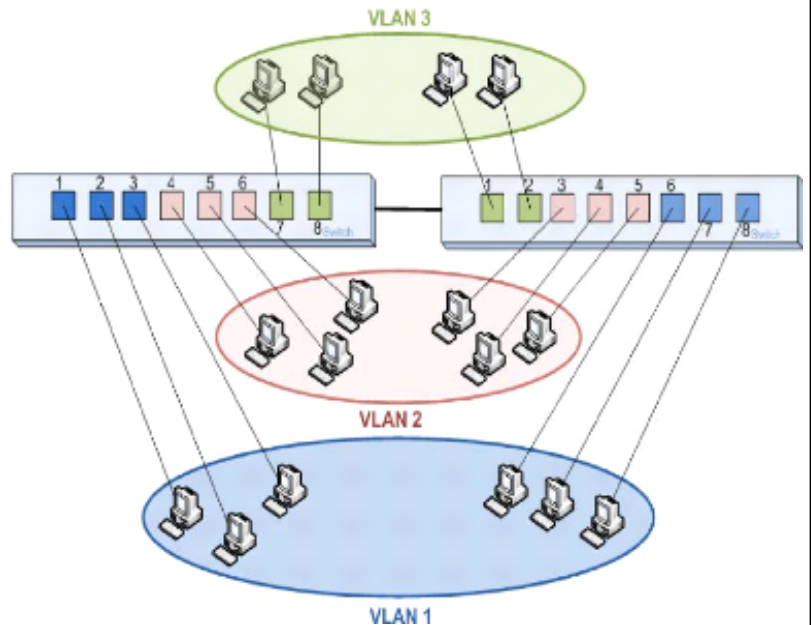
- Couche 2 : Tous les hôtes peuvent s'envoyer des trames car elles part
- Couche 3 : Tous les hôtes appartiennent au même sous-réseau IPV4, et peuvent s'envoyer des paquets sans passer par une passerelle

Une solution pour **limiter** les risques qu'un attaquant puisse opérer un MITM au sein d'un LAN est de **segmenter le LAN en plusieurs VLAN**.

Les VLANs permettent de "couper" un switch en plusieurs switches virtuels, permettant ainsi d'isoler des groupes d'hôtes entre eux

Les hôtes de chaque VLAN peuvent ainsi échanger des trames entre eux, mais pas avec des hôtes d'un autre VLAN (sauf par l'intermédiaire d'un routeur inter-VLAN)

Limitation : Cela n'empêche pas les attaques par ARP poisoning au sein même d'un VLAN !



4.2 Sur les commutateurs : Fonction DHCP Snooping et Deep ARP inspection

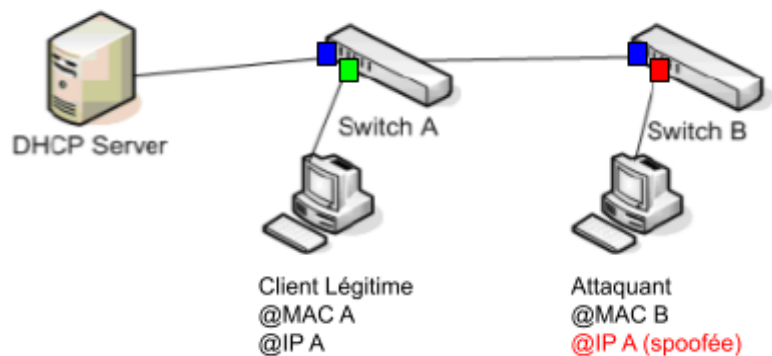
La fonction **DHCP snooping** (fouiner, espionner) installée sur un **commutateur** est normalement conçue pour empêcher un attaquant de se faire passer pour un serveur DHCP (on identifie alors les ports d'où les réponses DHCP sont censées venir), mais elle va également leur permettre d'enregistrer l'attribution des @IP aux différents hôtes et les associer à leur @MAC ainsi qu'au port auxquels ils sont connectés.

Lors d'une réponse à une requête DHCP, un commutateur pratiquant le DHCP snooping va retenir 3 choses :

- L'@IP attribuée à l'hôte
- L'@MAC de l'hôte
- Le port derrière lequel se situe l'hôte (qu'il y soit ou non directement connecté)

Par conséquent, le switch n'autorisera des réponses ARP associées à cette @IP qu'en provenance de ce port et de cette @MAC. Dans le cas contraire, le port duquel la réponse ARP usurpée provient sera désactivé

Limitation : Un attaquant pourra toujours utiliser des adresses IP non distribuées par DHCP (par exemple l'adresse IP statique d'un serveur Web et celle de sa BDD associée) et procéder au MITM entre ces deux serveurs



La réponse DHCP Offer du serveur est envoyée en broadcast

- Le Switch A sait que l'hôte IP-A/MAC-A est connecté à son port vert
- Le Switch B sait que l'hôte IP-A/MAC-A est quelque part derrière son port bleu

Si le switch B s'aperçoit qu'une réponse ARP émanant de l'IP-A est associée à une autre @MAC et provient d'un port non enregistré, il aura la possibilité de désactiver ce port.

4.3 Sur les hôtes : Mettre en place une table ARP statique

Sur un réseau peu évolutif, une autre solution peut consister à renseigner manuellement des couple MAC/IP dans la table d'un hôte.

On peut procéder de la sorte en utilisant des outils permettant de verrouiller les correspondances "appries", à un moment où l'on est sûr qu'il n'y a pas d'attaque. Le reste du temps, le système ne tient pas compte des réponses ARP qu'il reçoit.

On peut également filtrer toutes les réponses ARP qui n'ont pas de requêtes antécédentes dans les secondes qui précèdent (Firewall stateful sur client)

Limitation 1 : Maintenance très élevée car les tables doivent être mises à jour constamment (ajout, modif, suppression)

Limitation 2 : Cette technique n'est pas infallible si l'attaquant procède également à une attaque de type MAC spoofing

