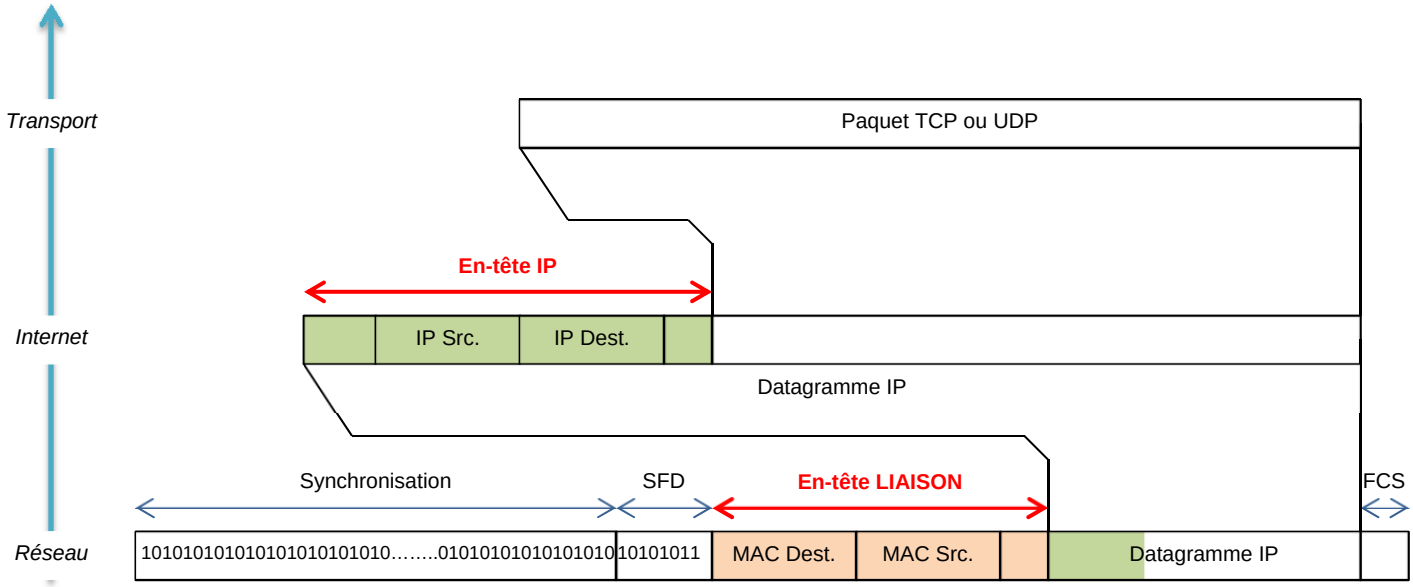
# Structure d'une trame Ethernet II

## 1 Structure générale.

C'est la trame que l'on rencontre dans la plupart des réseaux locaux actuels.

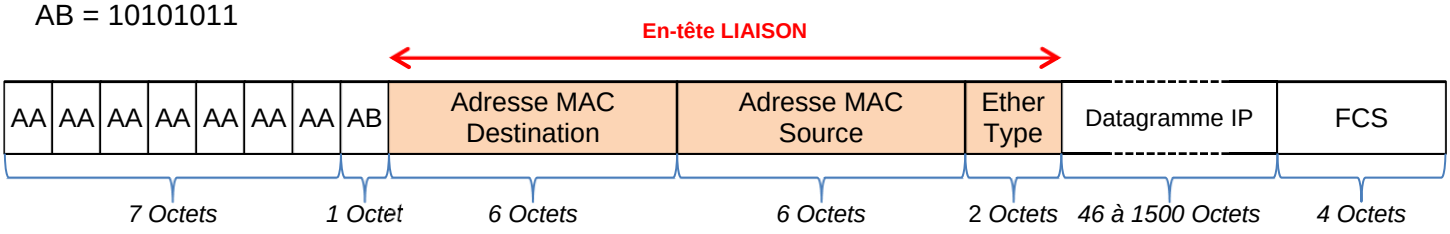
*Schéma de la structure*




Ce sont ces informations qui circulent sur le réseau

## 2 Trame ETHERNET II

AA = 10101010
AB = 10101011




**Préambule** : (7 octets) Permet la synchronisation des horloges de transmission. Il s'agit d'une suite de 1 et de 0 soit 7 octets à la valeur 0xAA

**SFD** : (1 octets) "Starting Frame Delimiter". Il s'agit d'un octet à la valeur 0xAB. Il doit être reçu en entier pour Valider le début de la trame.

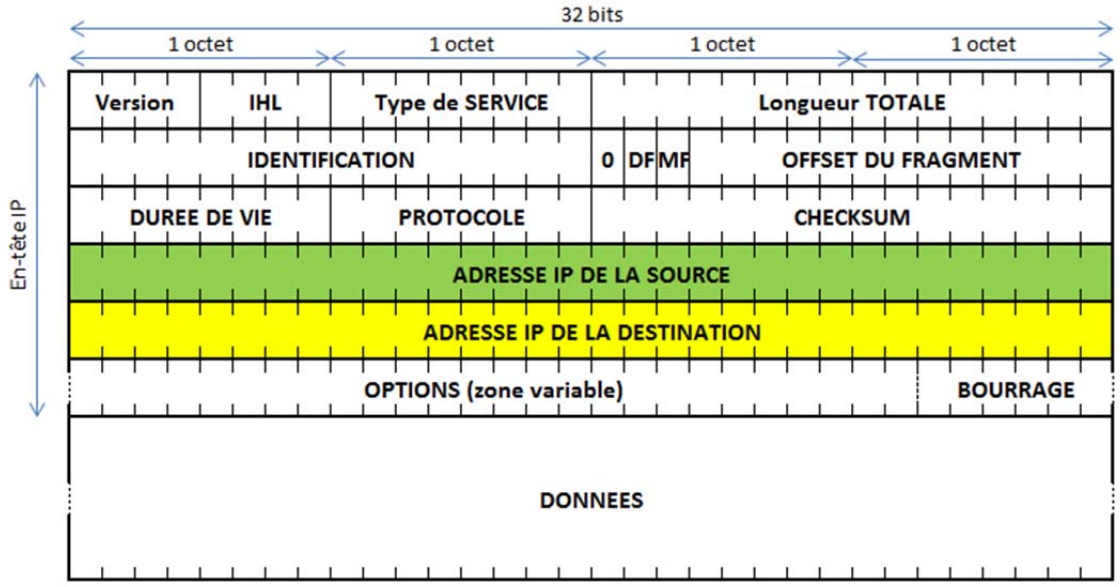
**En-tête** : (14 octets)
- Adresse MAC du destinataire (6 octets)
- Adresse MAC de l'émetteur (6 octets)
- EtherType (Type de protocole) (2 octets)

*Exemples de valeurs du champ EtherType* ➔

| EtherType | Protocole |
|---|---|
| 0x0800 | IPv4 |
| 0x0806 | ARP |
| 0x809B | AppleTalk |
| 0x8035 | RARP |
| 0x86DD | IPv6 |

**FCS** : (4 octets) Frame Check Sequence.
Ensemble d'octets permettant de vérifier que la réception s'est effectuée sans erreur.

# 3 Datagramme IP




**Version** : (4 bits) il indique le numéro de version du protocole IP utilisé (généralement 4).

**IHL** : (4 bits) Internet Header Lenght (Longueur d'entête). Spécifie la longueur de l'en-tête du Datagramme en nombre de mots de 32 bits. Ce champ ne peut prendre une valeur inférieure à 5.

**Type de service** : (8 bits) Donne une indication sur la qualité de « service » souhaitée pour l'acheminement des données.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Priorité | | | D | T | R | C | x |

| *Bits 0-2* | **Priorité** | 010➔Immédiate 001➔Normale 000➔Basse | |
|---|---|---|---|
| *Bit 3* | **D** | 0 = Retard standard | 1 = Retard faible |
| *Bit 4* | **T** | 0 = Débit standard | 1 = Haut débit |
| *Bit 5* | **R** | 0 = Taux d'erreur standard | 1 = Taux d'erreur faible |
| *Bit 6* | **C** | 0 = Coût standard | 1 = Coût faible |
| *Bit 7* | **x** | Réservé | |

**Longueur totale** : (16 bits) Longueur du datagramme entier y compris en-tête et données mesurée en octets.

**Identification** : (16 bits) Valeur assignée par l'émetteur pour identifier les fragments d'un même datagramme.

**Flags** : (3 bits) Commutateurs de contrôle :

- Bit 0 Réservé, doit être laissé à 0
- Bit 1 (DF - Don't fragment) 0= Fragmenté 1= Non fragmenté
- Bit 2 (MF – More Fragment) 0= Dernier fragment 1= Fragment

**OFFSET** : (13 bits) Décalage du premier octet du fragment par rapport au datagramme complet non fragmenté. Cette position est mesurée en blocs de 8 octets (64 bits).

**Durée de vie** : (8 bits) Temps en secondes pendant lequel le datagramme doit rester dans le réseau. Si ce champ vaut 0, le datagramme doit être détruit. Ce temps diminue à chaque passage du datagramme d'une machine à l'autre.

**Protocole** : (8 bits) Protocole porté par le datagramme (au-dessus de la couche IP)

| Valeur | Protocole |
|---|---|
| 1 | ICMP |
| 6 | TCP |
| 17 | UDP |
| Etc | etc |

**Checksum** : (16 bits) (Somme de contrôle) C'est une valeur qui permet de déceler une éventuelle erreur de transmission avec une très grande probabilité.

**IP Source** : (32 bits) Adresse IP de l'émetteur.

**IP Destination** : (32 bits) Adresse IP du destinataire.

**Options** : (Variable) Le champ est de longueur variable.Un datagramme peut comporter 0 ou plusieurs options.

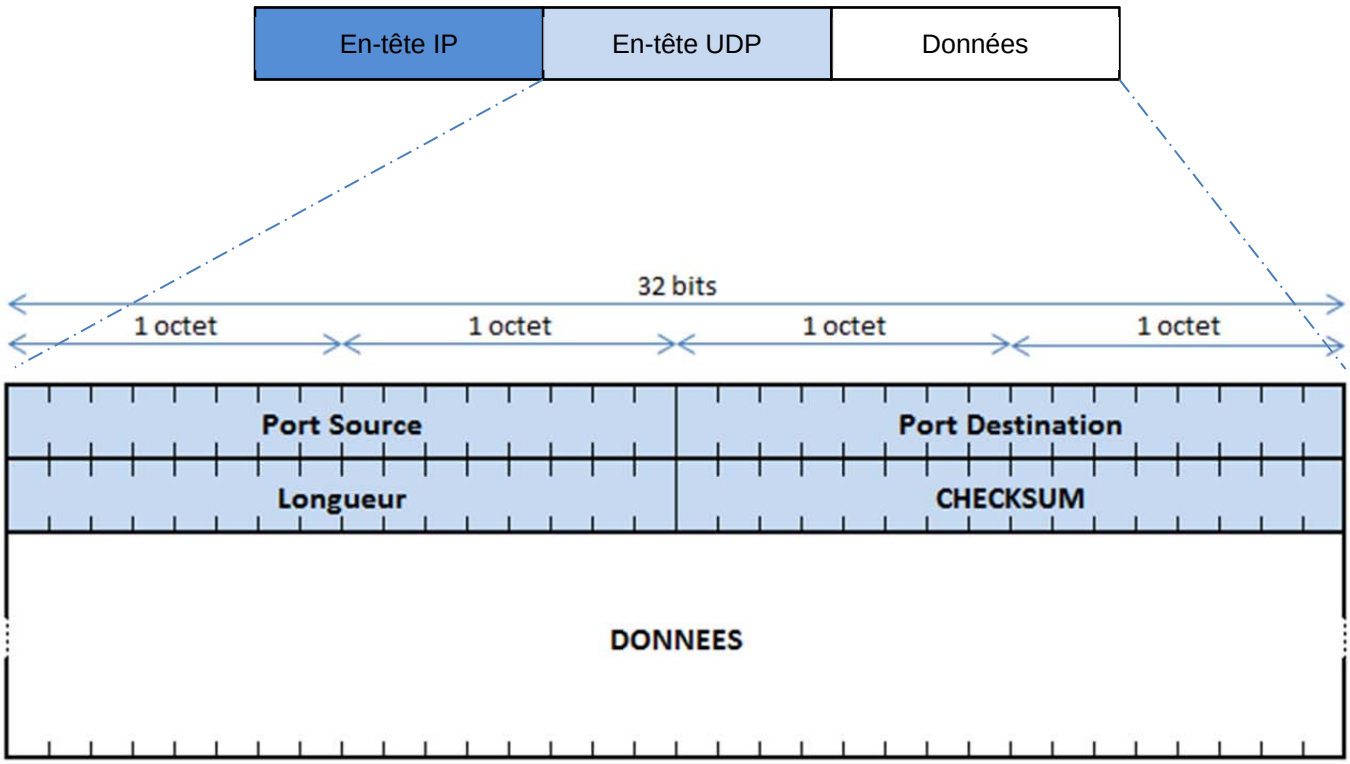
**Bourrage** : (Variable) Le champ Bourrage n'existe que pour assurer à l'en-tête une taille totale multiple de 4 octets. Le bourrage se fait par des octets à 0.

# 4 Datagramme UDP (User Datagram Protocol)

Le ***User Datagram Protocol*** (UDP, en français **protocole de datagramme utilisateur**) est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport de la pile de protocole TCP/IP.
Le rôle de ce protocole est de permettre la transmission de données de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port.

La nature de UDP le rend utile pour transmettre rapidement de petites quantités de données, depuis un serveur vers de nombreux clients ou bien dans des cas où la perte d'un datagramme est moins gênante que l'attente de sa retransmission (la voix sur IP, les jeux en ligne,…).

Le paquet UDP est encapsulé dans un paquet IP. Il comporte un en-tête suivi des données proprement dites à transporter.

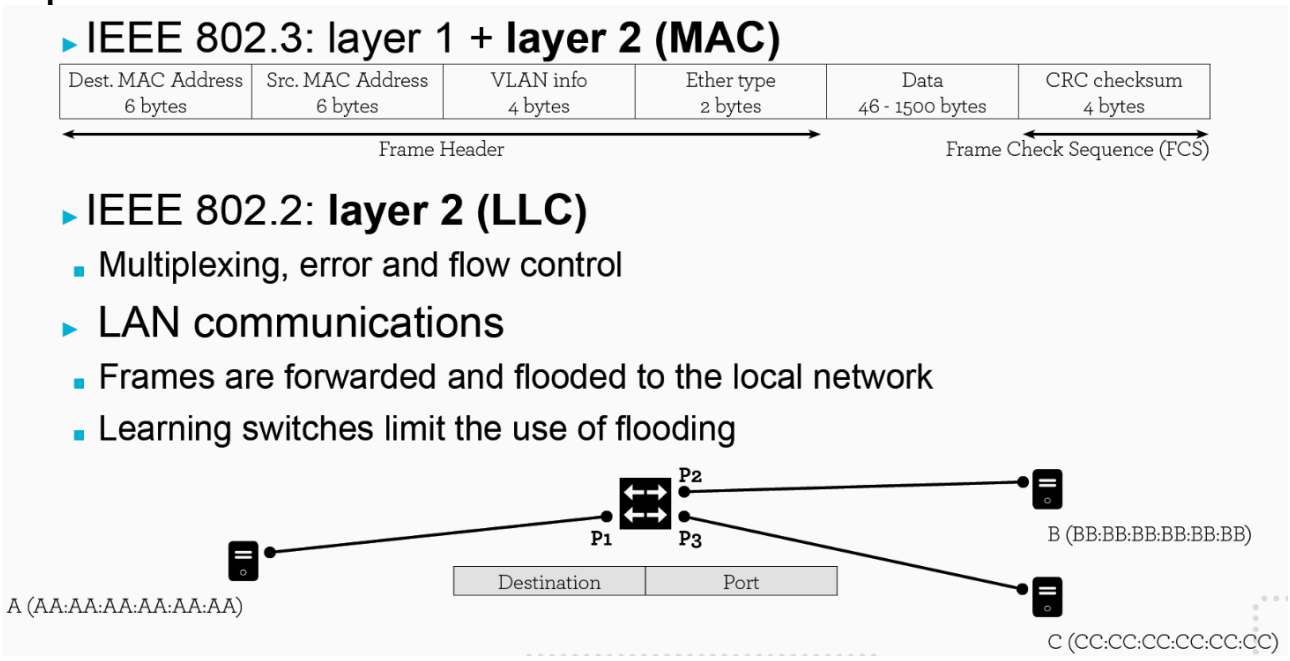


**S1.L1**

**Menaces sur la couche liaison**

L'objectif de ce cours est d'une part de connaître les vulnérabilités associées à la couche liaison et de comprendre comment celles-ci peuvent être exploitées pour mener une attaque impactant les fonctionnalités du réseau et ainsi les machines qui y sont connectées.

**Le protocole Ethernet**




Nous revoyons tout d'abord le protocole Ethernet. Ethernet est défini au sein des normes IEEE 802.3 et IEEE 802.2. C'est un protocole qui est à cheval entre la couche physique (ou couche 1) et la couche liaison de données (ou couche 2). Ici nous allons nous focaliser sur la couche 2 uniquement. On distingue en fait deux sous-couches : MAC pour Media Access Control et LLC pour Logical Link Control.

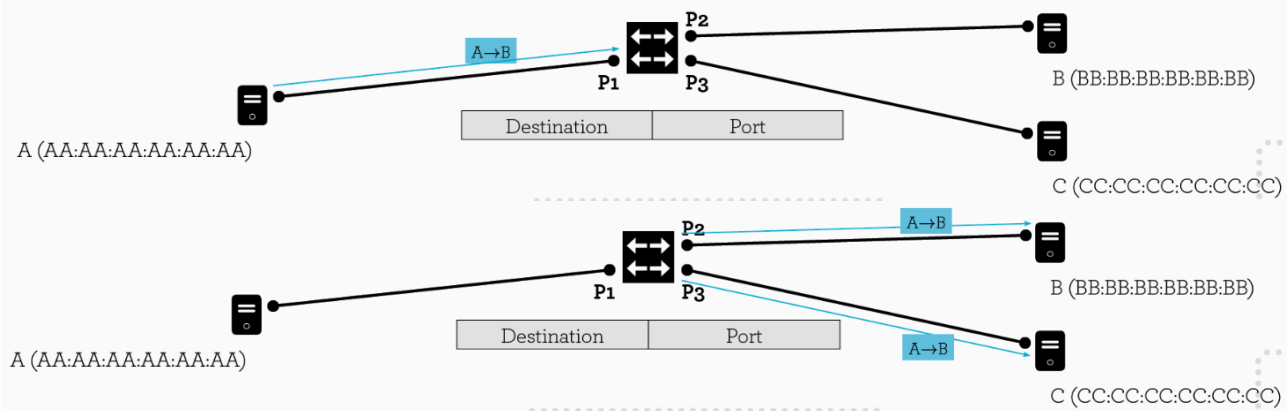
Le rôle principal de la sous-couche MAC est d'assurer le codage et le décodage des informations sous formes de trames de manière compatible avec le support physique sous-jacent. De plus, on retrouve au niveau de ces trames l'adressage dit Ethernet, MAC ou physique qui permet de distinguer sous un même réseau des machines distinctes. Au niveau de l'entête de la couche MAC dans une trame Ethernet, on retrouve notamment une indication sur le type de protocole de niveau supérieur utilisé

dans la partie données, par exemple IPv4 est encodé par la valeur 0x0800 en hexadécimal alors qu'IPv6 a comme code 0x86DD en hexadecimal. Pour information, toute valeur inférieure à 1500 indique la longueur des données de la trame dans une version moins courante d'Ethernet.

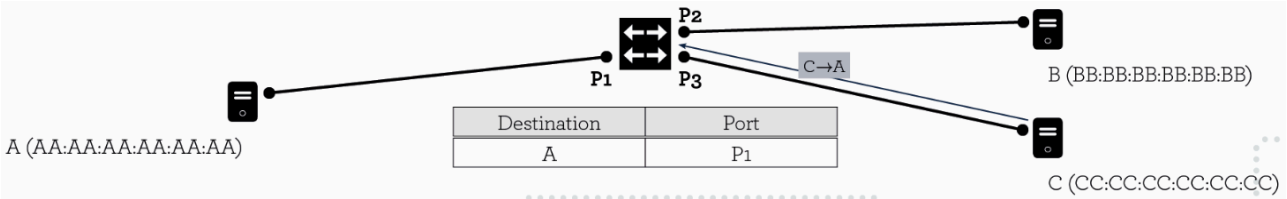
Nous distinguons ici également un champ dédié aux informations de VLAN. Ce dernier est optionnel et est aussi appelé champs IEEE 802.1Q, faisant référence à une extension de la norme. On y retrouve notamment un identifiant de réseau virtuel et la priorité associée. Enfin on retrouve un code de redondance cyclique ou CRC calculé sur l'ensemble de la trame, hors CRC bien entendu, ceci permettant de détecter toute erreur. Un point important est la taille des données qui peuvent être encapsulées. La taille maximale est le MTU (Maximum Transmission Unit) fixé à 1500 octets dans les réseaux Ethernet pour des raisons de fiabilité bien que des évolutions ont été proposées,comme les jumbo frames, permettant de transférer plus de données simultanément dans les réseaux Gigabits. La taille minimale d'une trame est de 64 octets. On en déduit que la partie données est de 46 ou 42 octets si le champs VLAN est présent ou non. Ainsi pour transférer moins d'informations, des octets seront rajoutés sous formes de padding, charge à la couche réseau de les supprimer ensuite.

La sous-couche LLC supporte des fonctionnalités avancées de contrôle de flux et d'erreurs, comme par exemple pour présenter les trames à la couche réseau dans le bon ordre. Cependant, Ethernet (802.3) n'utilise pas ces fonctionnalités qui sont déléguées aux protocoles de couches supérieures telles que la couche transport où TCP assure le réordonnancèrent des segments. Dans ce contexte, la couche LLC ne sert que d'interface entre la couche MAC et la couche 3, notamment pour s'assurer de la bonne division en trames (multiplexage), et dans certains cas elle n'est pas du tout utilisée.

Le protocole Ethernet assure la transmission des paquets au niveau d'un réseau local. De base, l'ensemble des trames sont simplement envoyées sur le médium physique et donc diffusées à tout le monde. Bien sûr ce modèle est très peu efficace et l'utilisation de commutateurs (ou switch) à la place de concentrateurs (ou hub) permet d'éviter de surcharger l'ensemble des liens avec les broadcasts.




Dans l'exemple ici avec les machines A, B et C, lorsque A envoie une trame à B, celle-ci est broadcastée sur tous les autres ports du switch car le switch n'a aucune information sur la localisation de B.




Par contre le switch sait maintenant que A est connectée sur le portP1. Lorsque C cherche à contacter A, la trame est alors transmise sur le port P1 uniquement.

**Les menaces portant sur la couche MAC**

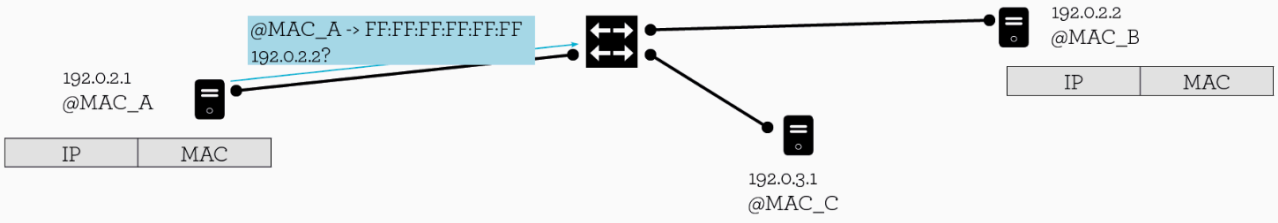
- MAC Headers can be easily forged → MAC address spoofing
  - Arbitrary source IP address = spoofing of any L2 host identity
  - In competition with the real host
- MAC Flooding:
  1. Attacker floods the switches using different source MAC addresses
  2. MAC Table of the switch is full → switch = hub
  3. Attacker can capture all the traffic

L'entête MAC n'est protégée par aucun mécanisme et peut dont être manipulée à volonté. Pour un attaquant il est ainsi facile d'émettre une trame avec une adresse source quelconque et de pouvoir se faire passer potentiellement pour une machine réellement existante sur le réseau. Ce faisant, les switchs vont associer à l'adresse MAC usurpée le port sur lequel se trouve l'attaquant, et non la machine légitime. Bien entendu, l'attaquant entre en conflit avec cette dernière dont il usurpe l'adresse. Il s'agit donc pour l'attaquant de maintenir l'association en envoyant de façon continuelle des trames.

Un autre avantage qu'un attaquant peut avoir à falsifer des adresses source est d'en générer un grand nombre. En particulier, un switch recevant de nombreuses trames portant des adresses MAC différentes devra alors conserver un très grand nombre de nouvelles associations port -- adresse MAC, susceptible d'excéder la taille de la table de commutation. En principe un switch n'arrête cependant pas totalement de fonctionner mais entre dans un mode dégradé, le mode hub. Dans ce mode, le switch broadcast simplement chaque trame entrante sur l'ensemble des autres ports. En effet, le switch n'a alors que pour seul choix de diffuser sur l'ensemble des ports car ne sachant pas exactement où la destination se trouve puisque la table de commutation est pleine et ne peut donc apprendre les nouvelles associations port – adresse MAC. Cette attaque peut être qualifée de déni de service car le switch ne fournit plus le service original. Cela permet également à un attaquant d'inonder l'ensemble des machines d'un réseau local mais également de capturer tout le trafic local. La confidentialité des échanges peut donc être compromise.

**Les menaces portant sur le protocole ARP**

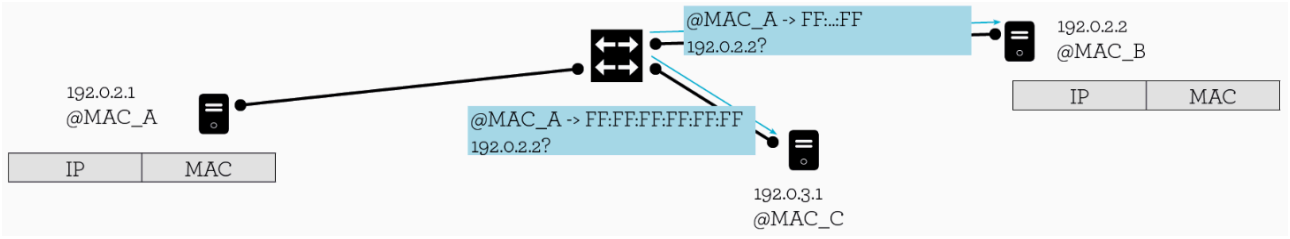
- MAC addresses pre-defined by hardware vendors
- Identification and connection to remote host:
  - .. → Hostname → **IP address** → **MAC address**
  - ARP = Address Resolution protocol (RFC 826)



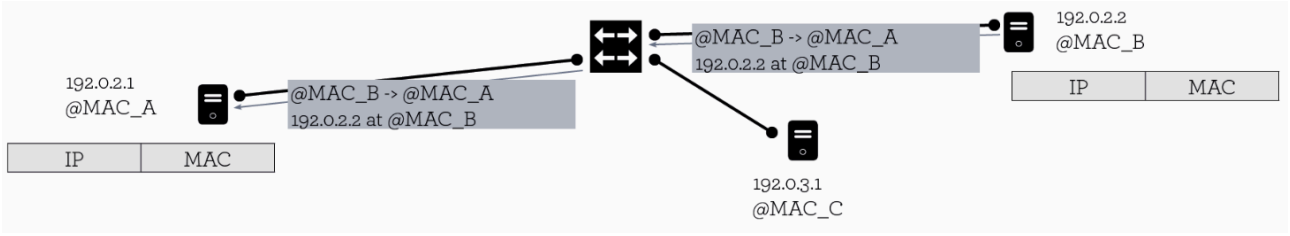

Les adresses MAC sont pré-définies par les fabricants. Cependant, une machine ne contacte pas directement une autre par son adresse MAC mais passe par différents mécanismes de résolution. Par exemple, un utilisateur va spécifier le nom d'une machine qui sera d'abord transformée en une adresse

IP grâce au service de DNS et enfin au niveau du réseau local grâce au protocole ARP qui permet d'associer une adresse MAC à une adresse IP. Nous voyons un exemple ici. L'hôte 192.0.2.1 cherche à contacter 192.0.2.2. On remarque sa table d'association adresse MAC - adresse IP est vide, elle ne peut donc pas savoir à qui transmettre le paquet.

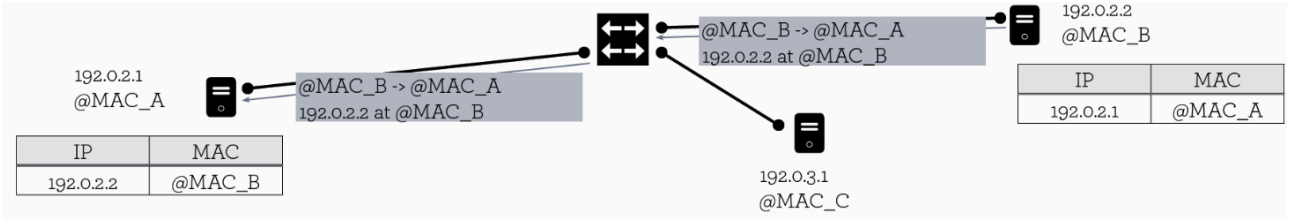
Elle envoie alors une requête ARP en broadcast avec l'adresse de destination FF:FF…




Toutes les machines du réseau local reçoivent cette requête et la machine cible reconnaît son adresse IP.

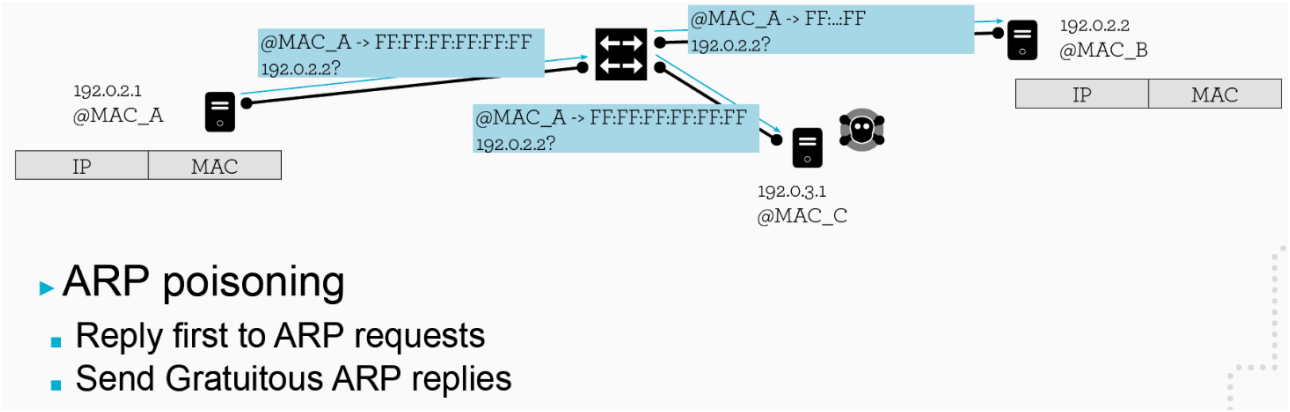



Elle y répond avec une réponse ARP en unicast cette fois à destination de 192.0.2.1

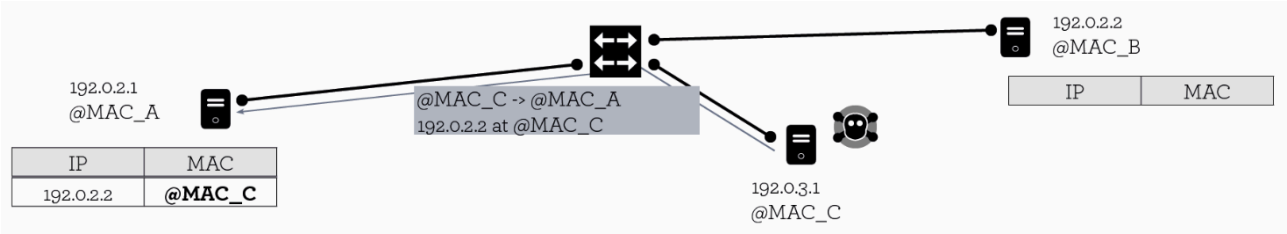



La table d'association MAC –IP de cette dernière est alors mise à jour et sera utilisée pour envoyer les paquets suivants à 192.0.2.2

Petite subtilité qui n'apparaît pas sur le schéma, la requête ARP contient également l'adresse IP source ce qui permet donc à 192.0.2.2 de mettre également sa table à jour.

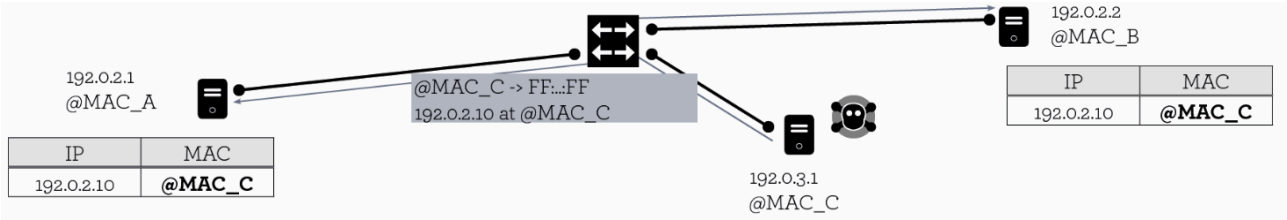



Nous allons voir maintenant comment le protocole peut facilement être détourné, en reprenant l'exemple précédent avec une requête émise par la machine à gauche cherchant à identifier l'adresse MAC de 192.0.2.2.



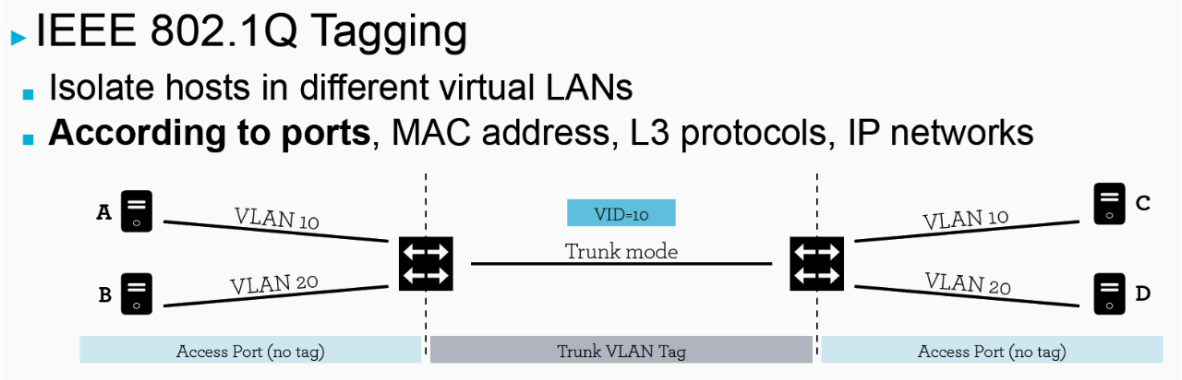

La machine avec l'adresse MAC_C, l'attaquant, répond en premier et falsifie la table ARP de la machine 192.0.2.1. Il est primordial que l'attaquant réponde en premier car c'est en principe la première réponse qui sera mise en cache pendant un temps déterminé et ce même si la machine 192.0.2.2 répond légitimement à la sollicitation ensuite.

Répondre en premier n'est pas le plus facile et un attaquant préfèrera utiliser le mécanisme de gratuitous ARP pour pallier cette difficulté.




Ce mécanisme permet à une machine de s'auto-identifier en envoyant l'équivalent d'une réponse ARP sans sollicitation préalable. C'est ce que fait l'attaquant ici en s'annonçant comme ayant l'adresse IP 192.0.2.10. Ce message ARP est diffusé en broadcast et affecte donc toutes les machines du réseau local. En imaginant maintenant que l'attaquant utilise plusieurs adresses IP, il peut alors totalement intercepter les communications locales et ainsi réaliser une attaque de type déni de service ou man-in-the-middle.

**Menaces portant sur les VLANs**




Nous avons précédemment évoqué un tag VLAN dans les trames. L'utilisation de VLAN ou de réseaux virtuels permet d'isoler des hôtes, dans ce cas précis au niveau 2. Ainsi deux hôtes dans différents VLANs ne pourront pas communiquer ensemble au niveau 2, ils devront forcément passer par une interconnexion de niveau 3 via une passerelle. L'utilisation de VLANs permet ainsi de faciliter la gestion des contrôles d'accès, par exemple il est possible d'interdire toute communication entre 2 VLANs même s'ils sont interconnectés par des passerelles de niveau 3.

Il existe plusieurs moyens de définir les trames appartenant à un VLAN, par exemple selon l'adresse MAC des hôtes, le protocole de niveau 3 utilisé ou le sous-réseau IP. Ici nous illustrons le cas classique qui définit un VLAN par rapport aux ports du switch. Deux VLANs sont configurés sur la figure présentée, le VLAN d'id 10 et le VLAN d'id 20, chacun avec deux hôtes A et C, et B et D respectivement. Ainsi lorsque la machine A enverra une trame à D, elle envoie la trame sur le lien qui la connecte au premier switch.

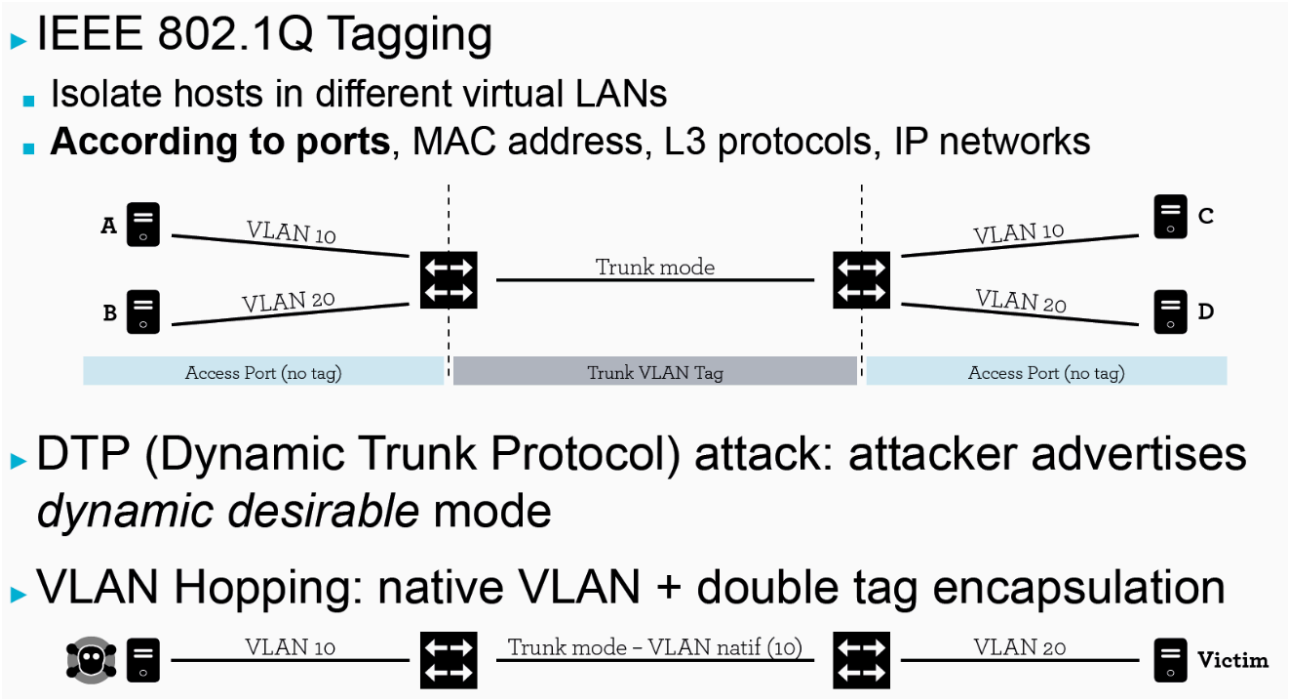
Pour elle rien ne change car ce sont les switchs qui vont gérer la partie VLAN. En effet, le lien concerné étant configuré en « access port » et identifié comme faisant partie du VLAN10, c'est le premier switch qui rajoute cet id dans l'entête avant de transmettre la trame au second switch. Ainsi, le second switch sera capable de déterminer de quel VLAN il s'agit et de l'envoyer sur le lien concerné.

Ici on enlève le tag avec l'id du VLAN. Certains liens locaux vont devoir en effet transmettre des trames de différents VLANs, ces liens sont dans un mode « trunk ».

Regardons maintenant les attaques et tout d'abord avec le protocole DTP.

Page 32 :

L'utilisation de VLANs requiert de configurer a priori chaque switch pour identifier les liens associés à des VLANs ainsi que le liens de type trunk qui véhiculeront les trames de plusieurs VLANs. Alors que la définition des VLANs est obligatoire, la définition des liens trunk peut se faire de manière automatisée grâce au protocole DTP ou Dynamic Trunk Protcol. Grâce à ce protocole, les switchs peuvent annoncer sur certains liens la capacité d'être trunk ou non, et plus précisément s'ils requièrent cette fonctionnalité ou s'ils peuvent la supporter si le switch de l'autre côté du lien en a besoin. Dans le premier cas on parle du mode « dynamic desirable » et dans le second du mode « dynamic auto ». Un attaquant peut donc se faire passer pour un switch et passer en mode « dynamic desirable », il négocie alors avec le switch avec lequel il est interconnecté et si celui-ci est compatible (desirable ou auto) alors le lien devient un lien trunk. L'attaquant pourra alors envoyer des trames à l'ensemble des VLANs même s'il n'était initialement associé qu'à un seul d'entre eux. Il pourra également intercepter les communications des différents VLANs, Différents types d'attaques s'ensuivent : man-in-the-middle, DDoS…




Voyons maintenant un autre type d'attaque qualifiée d'unidirectionnelle car elle permet uniquement à un attaquant d'envoyer une trame à destination d'une machine dans un VLAN pour lequel il n'a pas d'accès a priori.
Dans l'exemple ici, l'attaquant est dans le VLAN 10 et la victime dans le VLAN 20. L'attaquant peut profiter de deux fonctionnalités. Premièrement, les liens trunks disposent également d'un VLAN dit

natif, ici d'id 10, qui permet de véhiculer du trafic qui n'est pas destiné en principe à d'autres VLANS. Cependant si la configuration est mal effectuée, rien n'empêche d'assigner le même VLAN id à un sous-réseau local comme celui où l'attaquant est connecté.

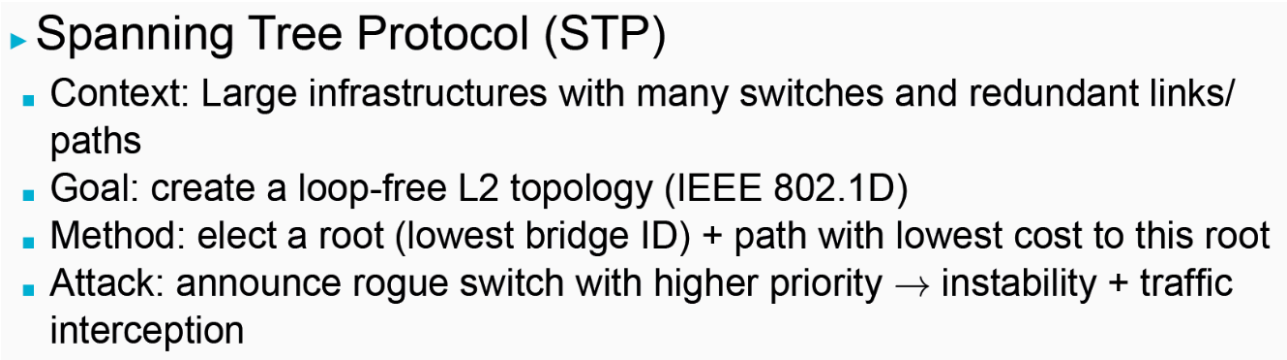

En principe, un switch ne jettera une trame entrante que si elle ne correspond pas au bon VLAN, ainsi l'attaquant peut émettre une trame avec le VLAN id à 10 mais en plus il encapsule une seconde fois avec le tag de VLAN correspondant à la victime soit 20.


Lorsque la trame arrive au niveau du premier switch, celle–ci doit être transmise sur le lien associé au VLAN 10. C'est donc équivalent au lien entre les deux switchs et le premier tag, du VLAN natif, est alors enlevé.


La trame continue d'être transmise et arrive au second switch avec le tag de VLAN id à 20. Ce switch reconnaît qu'il doit envoyer la trame vers la victime d'après son adresse MAC et que le lien à utiliser pour le faire est dans le VLAN 20. Cela correspond bien au VLAN ID de la trame ce dernier tag est enlevé et la trame est transmise à destination. L'attaquant a réussi à se positionner artificiellement dans le VLAN20 et à envoyer une trame dans ce dernier.

**Autres menaces**

Dans cette dernière partie, nous allons nous intéresser à deux protocoles principalement utilisés par les équipements réseaux et non par des machines utilisateurs.




Le premier STP ou Spanning Tree Protocol est un protocole essentiel permettant d'identifier automatiquement les chemins de niveau deux. Vous avez l'habitude de voir des exemples de topologie très bien formée en forme d'arbre ou étoile où vous distinguez facilement des chemins uniques entre deux nœuds quelconques. Imaginez maintenant une topologie beaucoup plus compliquée avec de nombreux chemins possibles entre deux switchs donnés. C'est en fait souvent le cas pour assurer une certaine redondance dans les chemins en cas de coupure d'un lien, ce qui introduit alors une boucle

de routage au niveau 2. Dans ce cas il faut définir bel et bien un chemin unique à un instant précis. Le protocole STP s'en charge en construisant un arbre de recouvrement au niveau logique au dessus d'une topologie quelconque.

Pour ce faire, un commutateur est élu comme racine de l'arbre, c'est en fait celui avec la plus petite priorité, cette dernière étant définie manuellement. En cas d'égalité l'adresse MAC permet de départager. Une fois le commutateur racine désigné, il devient alors possible pour les switchs connectés à ce commutateur de trouver une route directe. Ces derniers calculent également le coût associé à ces routes directes selon le type de liaison physique. Ensuite, par un effet de propagation, les coûts se cumulent sur les différents liens et chaque switch ou commutateur peut alors calculer le coût pour atteindre le commutateur racine à partir des différents ports de sortie et au final désigner celui qui a le plus petit coût comme le port à utiliser dit port racine.

Une attaque facilement réalisable est donc d'annoncer un commutateur avec une priorité volontairement très faible pour se faire élire comme commutateur racine et donc intercepter une grande partie du trafic puisque ce rôle de racine confère une position centrale dans l'arbre de recouvrement.

En variant régulièrement la priorité, et même en utilisant plusieurs faux commutateurs avec un tel comportement, l'algorithme recalcule de manière incessante un nouvel arbre de recouvrement qui perturbe la bonne stabilité de fonctionnement, c'est une attaque de déni de service.

- **Spanning Tree Protocol (STP)**
  - Context: Large infrastructures with many switches and redundant links/ paths
  - Goal: create a loop-free L2 topology (IEEE 802.1D)
  - Method: elect a root (lowest bridge ID) + path with lowest cost to this root
  - Attack: announce rogue switch with higher priority → instability + traffic interception
- **Link Layer Discovery Protocol (LLDP)**
  - Advertise information (name, vendor, version, addresses) to neighbors
  - Used for managing network, e.g. discovering topologies
  - No security of exchanged messages:
    - Eavesdropping: access sensitive information to prepare an attack
    - Information modification to make network management inefficient

LLDP est un autre protocole pour la configuration  et la gestion de réseaux. Ce protocole permet à chaque équipement connecté d'envoyer des informations à ses voisins sur sa configuration: adresses IP, type, nom, version… Ces informations sont notamment utiles pour un administrateur qui peut alors facilement reconstruire la topologie spécifique d'un réseau car toutes les informations reçues par un équipement sont sauvegardées dans une base locale. C'est également un protocole largement utilisé par les nouveaux réseaux, type SDN (Software-Defined Networking) qui permet justement aux différents switchs de découvrir ses voisins et ainsi reconstruire la topologie automatiquement au niveau d'un contrôleur SDN central.

On voit alors facilement les risques associés puisque ces informations sont échangées sans authentification ni chiffrement. Un attaquant peut donc également les récupérer et ainsi construire une base de connaissances sur le système qu'il cible. Ainsi, il pourra plus facilement élaborer une

attaque future, par exemple selon les vulnérabilités connues des équipements qui se sont auto identifiés avec LLDP.  L'attaquant peut également diffuser de fausses informations pour perturber la supervision du réseau.

C'est sur ces dernières menaces que se termine ce cours.

QUIZ : Menaces sur la couche LLC

1) Quel est l'intérêt principal pour un attaquant de forcer un switch sur lequel il est connecté à passer en mode hub ?

a) réaliser une attaque de déni de service tierce
b) intercepter l'ensemble des communications des machines connectées sur ce switch éviter d'être identifiable
c) usurper l'identité d'une autre machine

2) Pour usurper une adresse MAC, l'utilisation du protocole ARP est-elle requise ?
   a) Oui
   b) Non

3) Quelles sont les raisons pour un attaquant d'utiliser le mécanisme de "gratuitous arp" pour réaliser une attaque ARP poisoning *( une reponse)*
   a) usurper plusieurs adresses IP en une seule fois
   b) usurper une adresse IP de manière proactive
   c) forcer le switch à passer en mode hub

4) Quel est l'intérêt pour un attaquant de créer un lien trunk pour les VLANs( *une réponse)*
   a) créer ses propres identifiants de VLANs
   b) transmettre les trames de ses propres VLANs sans être filtré
   c) contrôler (au moins partiellement) les trames d'autres VLANs
   d) forcer la déconnexion des autres liens trunk

5) Quelles sont les propriétés qu'un attaquant exploite pour réaliser une attaque de type VLAN hopping ?( 2 reponses possibles)

a) L'utilisation possible d'un VLAN id natif réservé pour les liens trunk
b) La configuration d'un switch en mode "dynamic desirable"
c) La possibilité de spécifier plusieurs entêtes de VLANs dans les trames
d) L'usurpation de l'adresse IP d'une machine du VLAN cible