acceis

Gwenn Feunteun

gwenn@acceis.fr

Architecture & sécurité réseau

Les attaques orientées réseau

istic Informatique Électronique

acceis

Les attaques orientées réseau

Il existe plusieurs types d'attaques réseau, ayant des buts différents :

- Usurper une identité ;
- Porter atteinte à la disponibilité d'un service ;
- Récupérer, voire altérer, des informations.

De manière générale, une attaque « réseau » est une attaque ne dépassant pas le niveau 4 sur le modèle OSI. Sinon on parle plutôt d'attaque applicative

Usurpation d'identité

Un cas d'école

L’usurpation d’identité

Les attaques par ***IP Spoofing*** consistent à usurper une adresse IP lors de communications sur le réseau. Elles étaient efficaces à une époque ou des services comme *rsh* (permettant l’exécution d’une commande sur une machine distante) ou *rlogin* (pour ouvrir une session à distance) étaient courants.

Ces services se basaient sur le login de l’utilisateur et l’adresse IP du client pour **l’authentifier**. En usurpant une adresse IP, on usurpait donc une identité complète.

Mais comment ?

RSH

Un démon *rshd* écoute sur le port **TCP** 514. Pour exécuter une tâche à distance, l'utilisateur peut utiliser une commande comme celle-ci :

```
rsh -l remoteuser host.example.com "mkdir testdir"
```

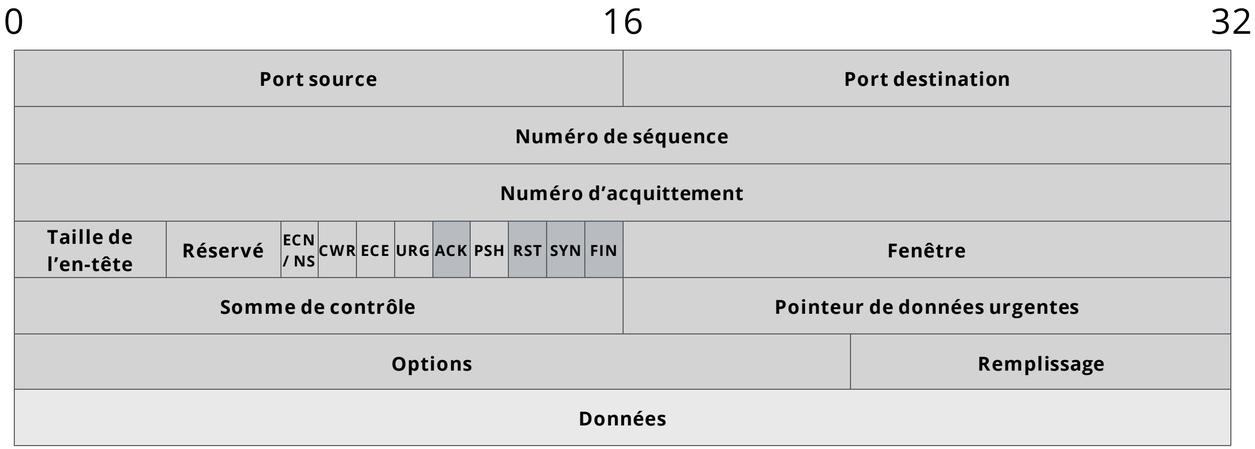
Toutefois, TCP fonctionne en mode connecté, avec contrôle de communication. Cela implique l'établissement d'une connexion au préalable, avec une série d'échanges bien spécifiques : le « *Three way handshake* »

Rappels sur TCP

Le protocole TCP (*Transmission Control Protocol*) se situe au dessus d'IP dans le modèle TCP/IP (couche transport dans le modèle OSI). C'est un protocole connecté (*statefull*) et fiable.

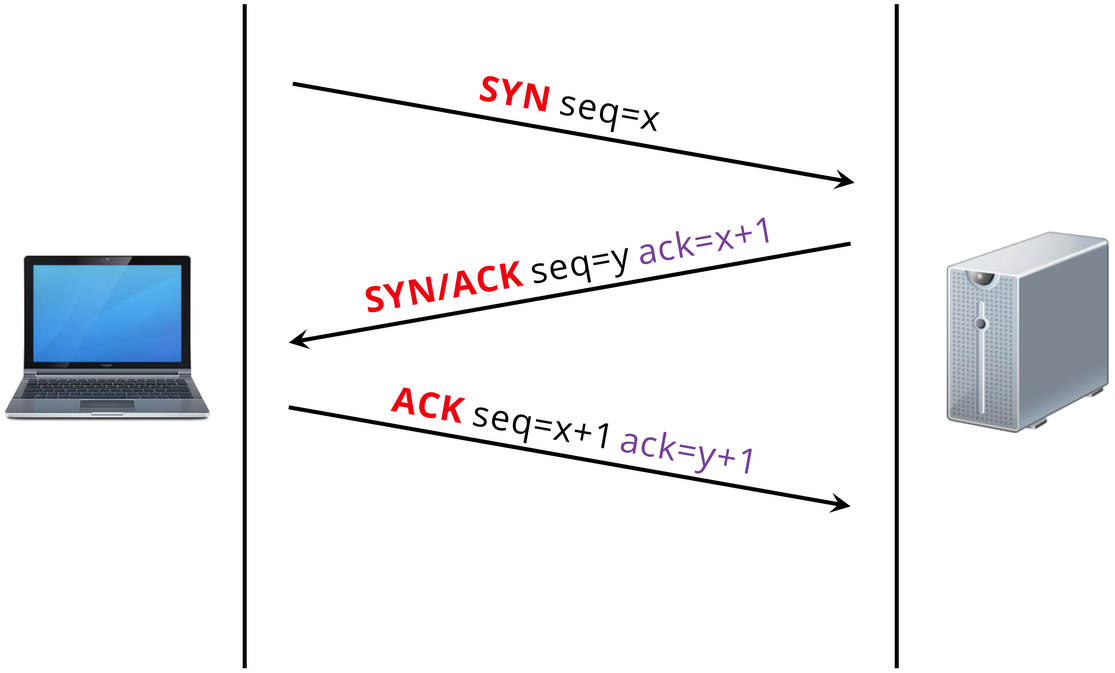
Une session TCP fonctionne en trois phases :

1. **Etablissement de la connexion ;**
2. **Transferts de données ;**
3. **Fin de la connexion.**



Rappels sur TCP

Etablissement d'une session



Afin de bien s'assurer que cette négociation s'effectue correctement, des correspondances sont établies entre les **numéros de séquence et d'acquittement.** Deux numéros sont donc générés :

- **x**, par l'initiateur de la connexion ;
- **y**, par le destinataire de la demande.

Ils sont ensuite incrémentés du nombre de données transmises dans le précédent segment envoyé.

IP Spoofing

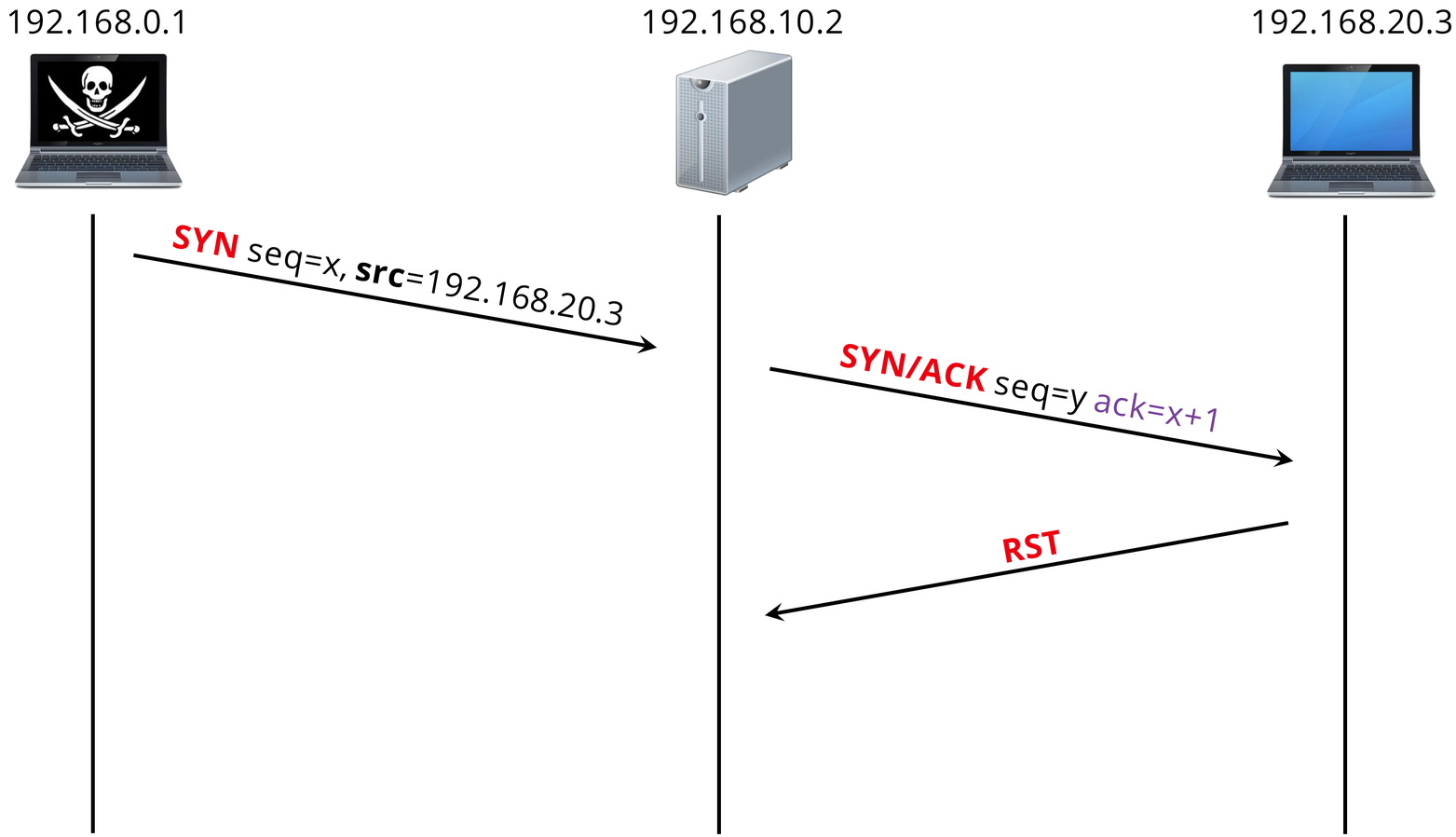
Le protocole IP n’est pas prévu pour validité de l’adresse source lors de l’envoi et du transit des datagrammes (c’est comme ça). Il est donc très facile de définir une **adresse source arbitraire**.

Problème : Les datagrammes sont envoyés avec comme adresse source l'adresse de la machine que l’on souhaite usurper, l'attaquant ne reçoit donc aucune réponse de la part du serveur : **il travaille en aveugle.**

Dans le cas de TCP, cela pose deux problèmes :

- L'attaquant ne peut pas connaître le nombre **X** envoyé par le serveur et ne peut donc pas envoyer le paquet ACK final ;
- La machine dont on usurpe l’identité va recevoir un paquet SYN/ACK non sollicité et va probablement répondre par un paquet RST qui va empêcher la connexion.

IP Spoofing



IP Spoofing

Si on veut usurper l'adresse IP d'une machine active sur le réseau, il faut l'empêcher de répondre RST au paquet SYN/ACK envoyé par le serveur. Il faut donc effectuer une attaque en déni de service, pour l'empêcher de communiquer sur le réseau. Il reste alors un problème : celui du **numéro de séquence Y** de la réponse du serveur, **que l'on ne connaît pas**.

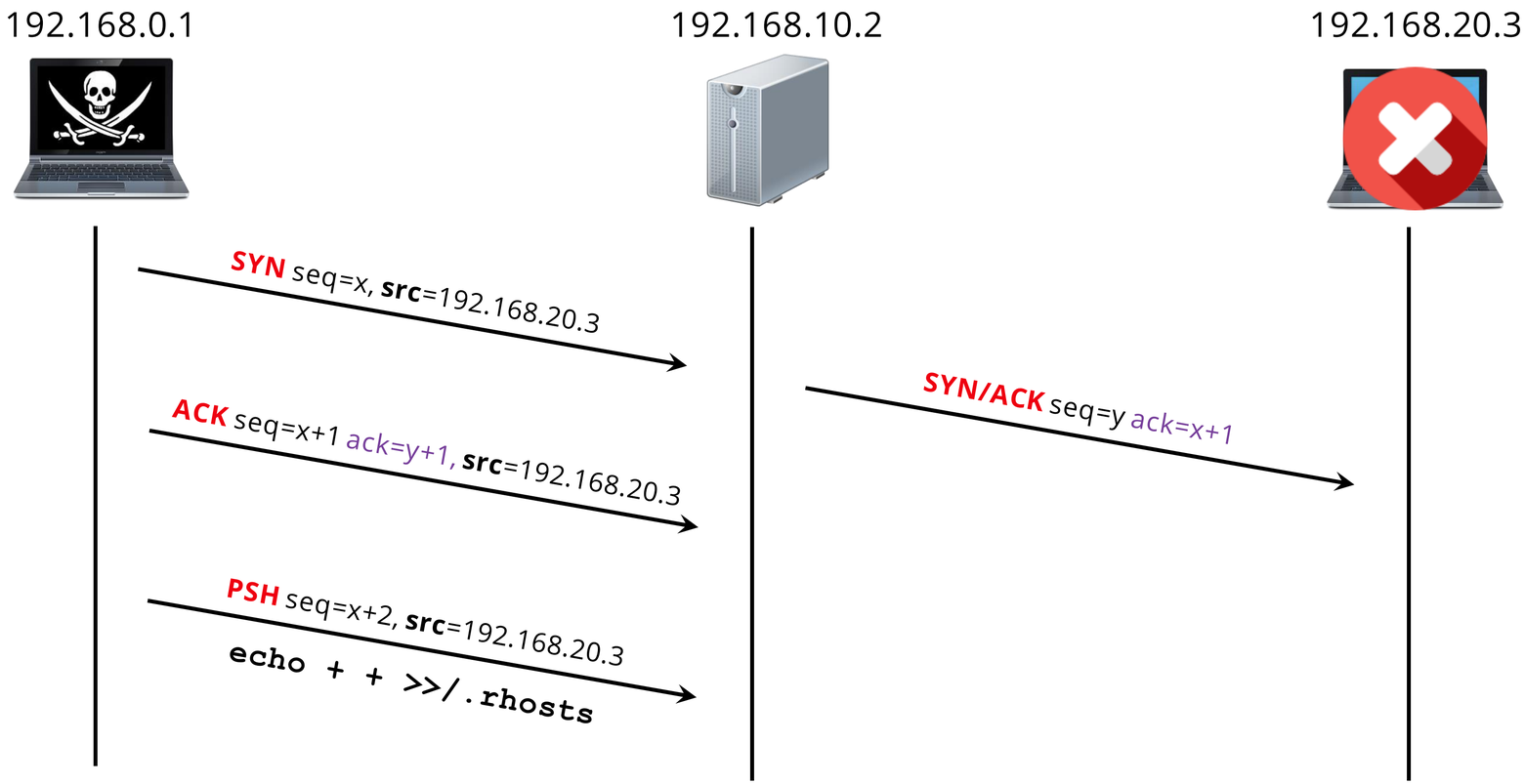
Deux solutions existent pour palier ce problème :

- Être en écoute sur le réseau pour intercepter le paquet provenant du serveur, afin de lire la valeur Y de son paquet SYN/ACK ;
- Parvenir à deviner sa valeur.

Les premières piles IP n'étaient pas suffisamment sécurisées et le numéro de séquence présentait une **entropie insuffisante**. Il était donc possible, en sollicitant la pile IP avant l'attaque, de déterminer une plage de valeurs possibles.

IP Spoofing

Cette technique a été la base de la très fameuse attaque de **Kevin Mitnick** sur **Tsutomu Shimomura**, à noël 1994.



IP Spoofing

Cette attaque est tombée en désuétude pour plusieurs raisons :

- Les piles IP ont été sécurisées, et l'ISN (*Initial Sequence Number*) a normalement une entropie suffisante pour qu'il ne soit plus devinable par un attaquant ;

- Les services qui utilisent des relations de confiance basées sur l'adresse IP ne sont plus utilisés (*rsh* a été remplacé par *SSH*).

Mais il est toujours possible (et facile) d'usurper une adresse IP source dans le cas d'un **protocole non connecté**, comme UDP par exemple.

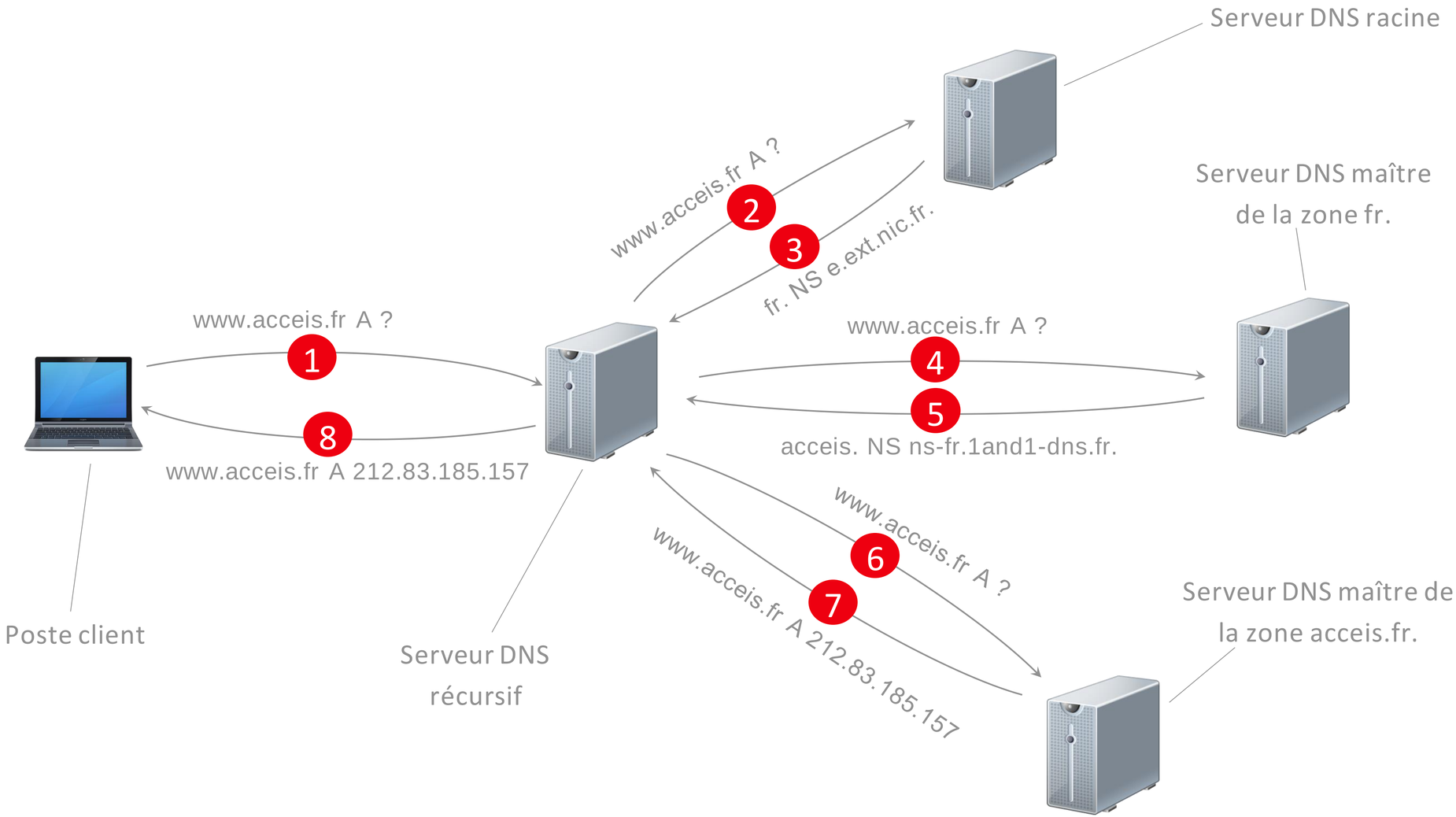
Le protocole DNS

Le *Domain Name System* (ou DNS, système de noms de domaine) est un service permettant de traduire un **nom de domaine** en **informations de plusieurs types qui y sont associées**, notamment en adresses IP de la machine portant ce nom.

Résoudre un nom de domaine consiste à trouver l'adresse IP qui lui est associée. Avant le DNS, la résolution d'un nom sur Internet devait se faire grâce à un fichier texte appelé HOSTS.TXT (RFC 6081) maintenu par le NIC du *Stanford Research Institute* (SRI) et recopié sur chaque ordinateur par transfert de fichier.

DNS utilise en général **UDP** et le **port 53**. La taille maximale des paquets utilisée est de 512 octets. Si une réponse dépasse cette taille, la norme prévoit que la requête doit être renvoyée sur le port TCP 53. Ce cas est cependant rare et évité (sauf parfois sous Windows), et les firewalls bloquent souvent le port TCP 53.

La résolution de nom

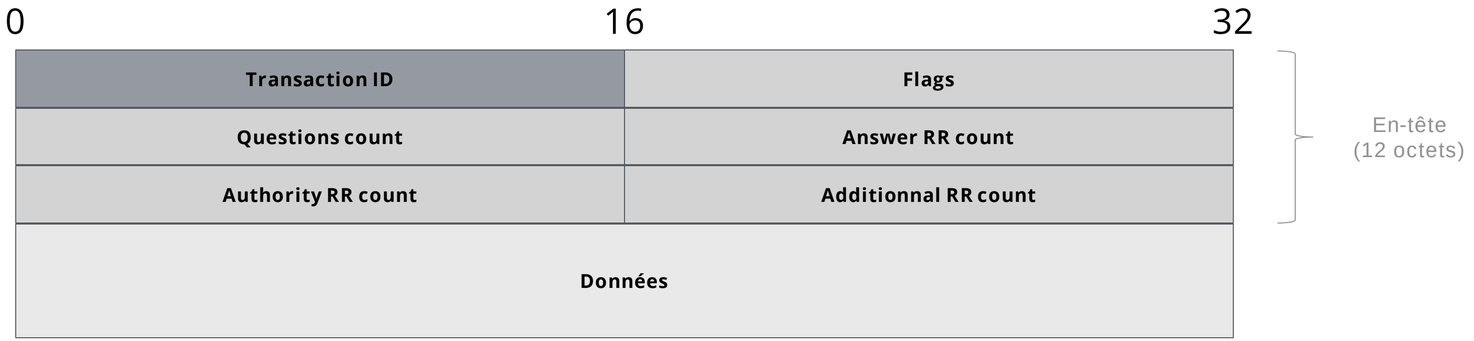


Le serveur cache

Afin d'améliorer les performances des serveurs récursifs et éviter la surcharge du réseau et des autres serveurs DNS, ils utilisent un **cache** pour stocker les réponses des requêtes. Pour les requêtes ultérieures des clients, le serveur consulte d'abord son cache pour voir si la requête n'avait pas déjà été traitée (par exemple pour un autre client). Si c'est le cas, les données correspondantes du cache sont fournies en réponse. Dans le cas contraire, le serveur lance le processus de résolution.

Les données du cache cependant une **durée de vie limitée** qui est spécifiée dans le champ TTL (Time To Live) associé à chaque ressource. Ceci permet de ne pas maintenir dans le cache des informations périmées et de les rafraîchir en fonction de la valeur du TTL.

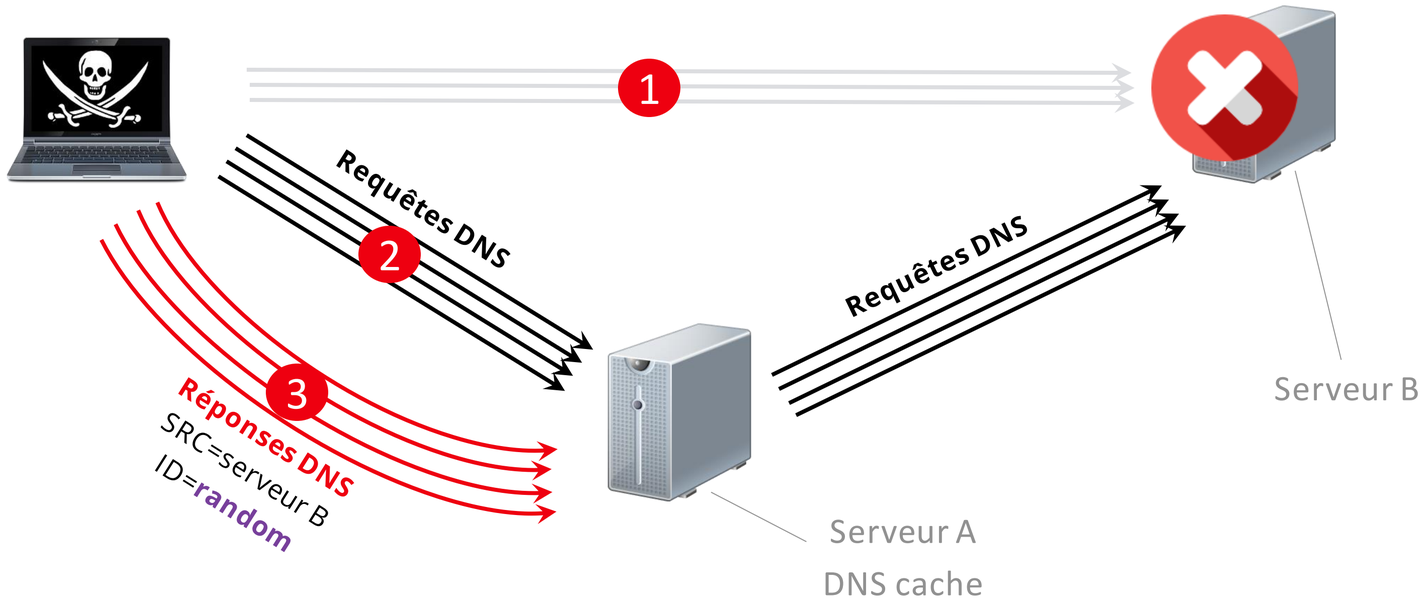
Format d’un datagramme



Le *Transaction ID* est un nombre entier représentant une requête (recopié lors de la réponse pour faire l’association avec la demande initiale), car UDP est *stateless*. Il sert à « authentifier » les requêtes.

Ce faible niveau de sécurité expose le protocole à des attaques par empoisonnement : il s’agit de leurrer les clients ou les serveurs DNS récursif afin de leur faire croire qu'ils reçoivent une requête valide tandis qu'elle est frauduleuse.

DNS Cache poisoning



Pré-requis :

- Connaître l'adresse IP du serveur B (facile) ;
- L'empêcher de répondre (plus compliqué) ;
- Connaître le **port source** de la requête DNS (certains serveurs utilisent toujours le même) ;
- Connaître l'**ID de transaction** utilisé (en théorie très difficile, mais moins grâce au paradoxe des annivervaires).

Paradoxe des anniversaires

Le paradoxe des anniversaires est une estimation probabiliste du nombre de personnes que l'on doit réunir pour avoir une chance sur deux que deux personnes de ce groupe aient leur anniversaire le même jour. Il se trouve que ce nombre est 23, ce qui choque un peu l'intuition. À partir d'un groupe de 57 personnes, la probabilité est supérieure à 99 %.

$$p(n) = 1 - \frac{|E|!}{(|E| - n)!} \cdot \frac{1}{|E|^n}$$

n	*p*(*n*)
5	2,71 %
10	11,69 %
15	25,29 %
20	41,14 %
23	50,73 %
25	56,87 %
30	70,63 %
40	89,12 %
50	97,04 %
60	99,41 %
80	99,99 %
100	99,99997 %

Déni de service

Déni de service

Un déni de service (en anglais ***Denial of Service*** ou ***DoS***) est un état dans lequel ledit service n'est plus en mesure d'être rendu, soit parce qu'il est dans l'incapacité d'assurer les fonctions pour lesquelles il a été conçu ou que ses utilisateurs ne peuvent plus y accéder.

Afin de provoquer un déni de service, il est possible d'attaquer :

- **L'application** rendant le service ;
- **Le système** supportant l'application ;
- **Le lien réseau** permettant d'accéder au service (incluant les composants intermédiaires) ;

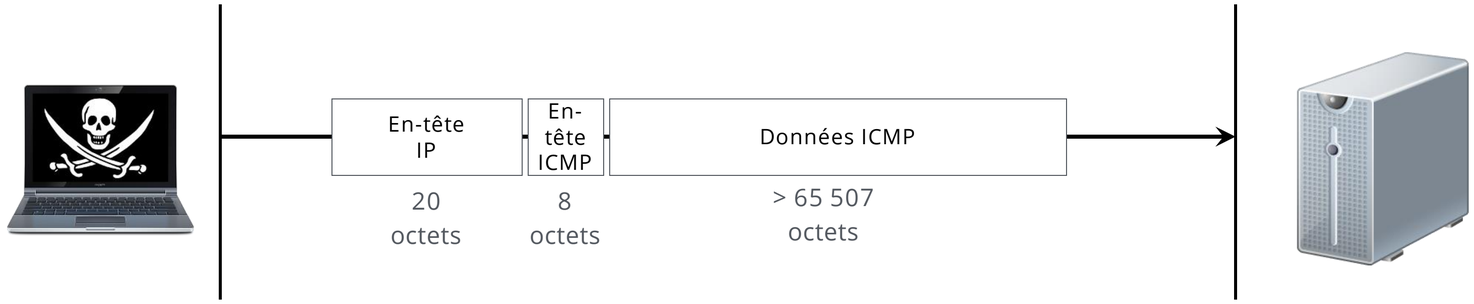
Mais aussi le matériel (rare), voire les utilisateurs (encore plus rare).

« Ping de la mort »

Le ping de la mort (aka *Ping of Death* ou *PoD*) est une vieille attaque des années 90 qui consistait à envoyer un paquet **ICMP ping malformé avec une taille supérieure à 65 535 octets**. La pile IP de certains systèmes et équipements (routeurs et imprimantes) n'arrivait pas à les gérer correctement au moment de rassembler le datagramme et adoptait un comportement hasardeux, voire plantait.

RFC 791

> Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets. Such long datagrams are



« Ping de la mort »

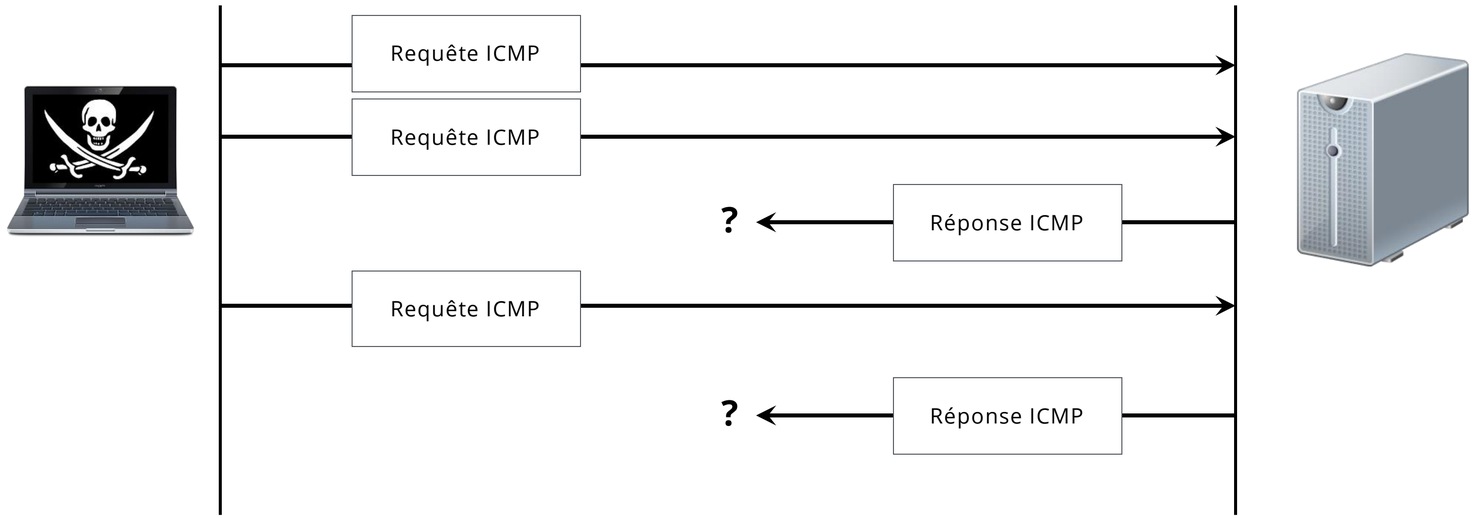
L'attaque est facile à réaliser car l'attaquant peut usurper (*spoofer*) très simplement l'adresse IP de l'émetteur.

Comment se protéger ?

- En utilisant des systèmes à jour ;
- En interdisant l'ICMP (solution pas très pertinente) ;
- En bloquant les datagrammes IP malformés (dont la taille est supérieure à 65 535 octets) au niveau d'un pare-feu ou d'un IPS pas exemple.

Ping flood

Le *ping flood* (ou *ICMP flood*) consiste à envoyer une très grande quantité de requêtes ICMP (en usurpant l'adresse IP source de préférence) afin de **saturer la bande passante de la cible** ou, si la cible est faiblement dimensionnée, **de consommer ses ressources** (RAM, CPU, etc). Généralement, c'est celui qui a le plus gros tuyau qui gagne. De plus, en répondant aux requête, la cible va également consommer de la bande passante en sortie.



Ping flood

Cette attaque était longtemps considérée comme désuète au regard des capacités des entreprises vs. celle d'un particulier, mais la démocratisation des connexions très haut débit la remet au goût du jour.

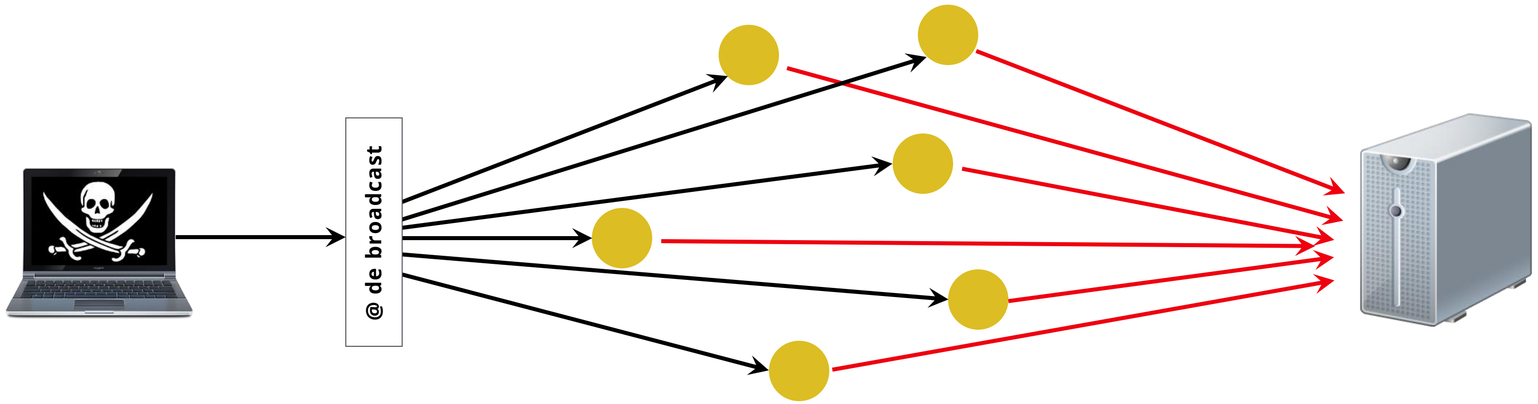
Comment se protéger ?

- En interdisant l'ICMP, mais l'attaque peut être réalisée avec d'autres protocoles ;
- En détectant les gros volumes de requête et en les limitant, voire en les bloquant, au travers d'un pare-feu ou d'un IPS, mais cela ne règle pas le problème de l'occupation de la bande passe en amont de l'équipement ;
- En utilisant des solutions commerciales conçues pour détecter ces attaques et « lisser » la charge sur ses infrastructures.

Attaque par réflexion

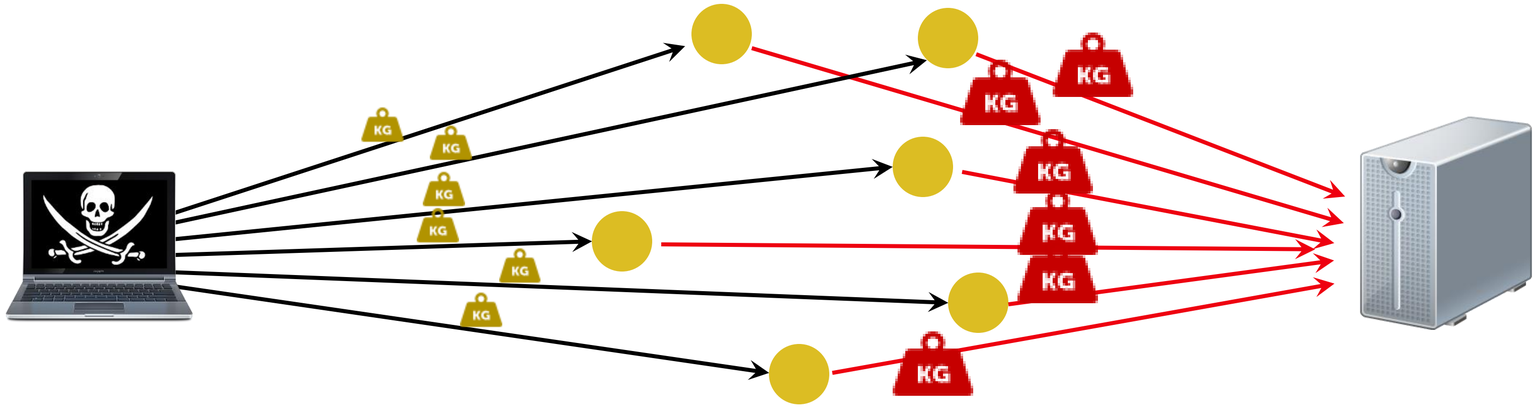
Les premières attaques par réflexion (*smurf attack* en anglais) datent des années 90. Elles consistaient à envoyer des requêtes ICMP (toujours en usurpant l'adresse IP source) à une adresse de *broadcast*, c'est-à-dire une adresse derrière laquelle se trouve plusieurs machines. Ces machines « intermédiaires » enverront leurs réponse à la cible, en saturant éventuellement sa bande passante.

Elles constituent des attaques en **déni de service distribué** (*Distibuted Denial of Service* ou *DDoS*), car plusieurs machines sont utilisées pour provoquer le déni de service.



Attaque par réflexion avec amplification

Les attaques par réflexion employant des requêtes ICMP avec des adresses de broadcast ne sont plus opérantes de nos jours. Il est toutefois possible d'utiliser une techniques similaire en exploitant **l'asymétrie entre les requêtes et les réponses de certains protocoles** (la réponse étant plus lourde en terme de taille et/ou de nombre de paquets).



Attaque par réflexion avec amplification

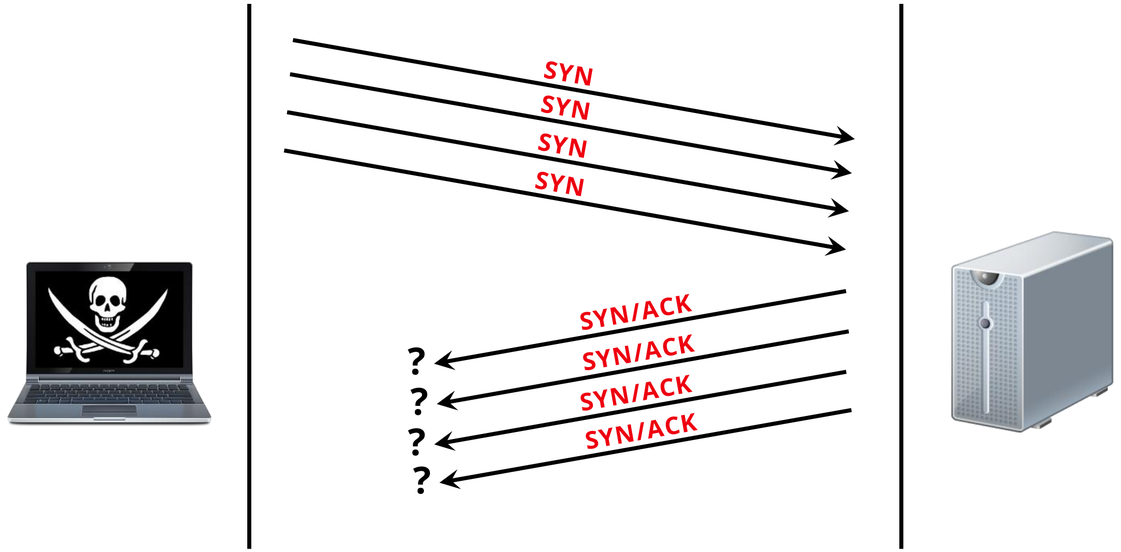
Les attaques par réflexion peuvent exploiter différents protocoles :

- NTP (*Network Time Protocol*) x556,9 ;
- DNS (*Domaine Name System*) x179 ;
- SNMP (*Simple Network Management Protocol*) x6,3 ;
- SSDP (*Simple Service Discovery Protocol*) x30,8 ;
- NBNS (*NetBIOS Name Service*) x3,8 ;
- Steam protocol x5,5 ;
- CHARGEN (*Character Generator protocol*) x358,8 **…**

Elles exploitent cependant systématiquement des protocoles ***stateless***, tels que ICMP ou UDP, afin d'usurper l'adresse IP source, condition *sine qua non* à leur réalisation.

SYN flood

Il s'agit d'initier un **grand nombre de connexions TCP**, mais que partiellement : l'attaquant ne répond pas au segment SYN/ACK. La connexion reste « **semi-ouverte** » coté serveur durant un certain délai (75 secondes lors des première attaques) avant d'être fermée. Chaque connexion consomme des ressources sur la machine cible (en particulier la file des connexions en attente d'établissement, qui a une taille limitée). En établissant un très grand nombre de connexions, il est possible de les épuiser.



SYN flood

Afin de laisser les slots occupés sur la machine cible, il est impératif de ne pas répondre à son SYN/ACK, sinon le slot sera libéré. Ce SYN/ACK va être reçu et traité par la pile IP du système de l'attaquant, qui va mettre fin à la demande de connexion en envoyant un RST, c'est le comportement normal d'une pile IP.

Plusieurs solutions peuvent permettre de gérer ce problème qui compromet l'attaque :

- Renseigner l'adresse IP source de chaque datagramme avec une valeur aléatoire. Les SYN/ACK sont envoyés à des hôtes distants inconnus/inexistants ;

- Empêcher le noyau de sa machine de répondre RST aux SYN/ACK, via une règle du pare-feu qui drop les RST à destination de la victime :

```
iptables -A OUTPUT -p tcp -s 192.168.0.1 -d 192.168.0.3 –tcp-flags RST RST -j DROP
```

SYN flood

Comment se protéger ?

- En limitant le nombre de connexions depuis la même source ou la même plage d'adresses IP ;
- En libérant les connexions semi-ouvertes dans un délai plus court ;
- En retardant l'envoi des réponses SYN/ACK en cas de surcharge ;
- En augmentant la taille de la file d'attente d'ouverture des connexions (paramètre *tcp_max_syn_backlog* sous Linux) ;
- En évitant d'allouer des ressources tant que la connexion n'est pas complètement établie, en utilisant par exemple des SYN cookies (vérification basée sur des numéros de séquence spécifiques).

Parfois c'est l'équipement en frontal (le pare-feu) qui ne supporte pas un très grand nombre de session concurrentes.

Exemples concrets

- **DYN**

Le 21 octobre 2016, environ 100 000 machines du botnet Mirai ont lancé une attaque contre les serveurs de DYN utilisés pour assurer la résolution de nom sur les domaines de leurs clients. Un grand nombre de services sont touchés : Twitter, Ebay, Netflix, GitHub, PayPal, sont inaccessibles pendant une dizaine d'heures. Le trafic généré a atteint 1 To/s.

- **Cedexis**

Le 10 mai 2017, une attaque DDoS de provenance inconnue a été menée sur les infrastructures de Cedexis. Beaucoup de sites ont été inaccessibles pendant l'après-midi (environ 200) et notamment : Le Monde, L'Équipe, Le Parisien, Top Achat et Rue du commerce.

Interception de communications

Interception de communications

L'objectif est ici de permettre à l'attaquant de prendre connaissance du contenu des échanges réseaux entre deux hôtes ou plus, voire d'altérer les données transmises. Plusieurs attaques peuvent être utilisées :

- Ecouter un média partagé (un hub ou un réseau WiFi par exemple) ;
- Utiliser un port miroir sur un switch ;
- Mettre une machine en coupure de flux physique ;
- Usurper l'identité des hôtes sur le réseau ;
- Détourner les flux de communication…

Rappels sur Ethernet II

Le protocole Ethernet se situe au dessous d'IP (couche liaison dans le modèle OSI, voire un peu physique). Il sert à assurer les communication entre les pairs d'un même sous-réseau ou plus particulièrement d'un même domaine de broadcast, c'est-à-dire sans passer par un routeur. Les hôtes sont identifiés par leur **adresse MAC** (*Media Access Control*), généralement de 6 octets, stockée au niveau de la carte réseau et théoriquement unique au monde.

Il y a quatre types de trame Ethernet :

- Ethernet originale version I (n'est plus utilisée) ;
- Ethernet Version 2 ou Ethernet II (toujours utilisée) ;
- IEEE 802.x LLC ;
- IEEE 802.x LLC/SNAP.

Ethernet et les réseaux commutés

Depuis sa création dans les années 80, Ethernet s'est progressivement développé, jusqu'à s'imposer comme le standard de communication de la couche liaison pour les réseaux locaux. Il fonctionne historiquement selon deux modes :

- Le **mode partagé**, où le support physique est commun entre les équipements du réseau ;
- Le **mode commuté**, où les terminaux sont reliés à un commutateur (un *switch*) en charge d'aiguiller les signaux à destination des bons interlocuteur.

Ethernet et les réseaux commutés

Le mode partagé

Le groupe de travail IEEE 802.3 a défini la technique d'accès au support physique (MAC pour *Medium Access Control*). L'objectif est notamment de gérer le problème des collisions (deux stations qui émettent en même temps). Pour cela, Ethernet utilise un mécanisme d'écoute du medium avant émission et de détection des collision : CSMA/CD.

De nombreux types de supports physiques peuvent être utilisés : câble coaxial, fibre optique, paire cuivrée, etc.

Cablâge 10BASE2

Hub Ethernet

Ethernet et les réseaux commutés

Le mode partagé

Dans ce mode, par conception, l'ensemble du trafic réseau est accessible à toutes les stations. Ce sont elles qui font le « tri » des trames Ethernet qui les concernent ou non, en fonction des adresses MAC.

Même si les HUB Ethernet permettent de raccorder chaque station avec un câble dédié connecté en RJ45, il n'y pas de segmentation des échanges : l'intégralité des flux est dupliquée sur tous les ports.

En basculant sa carte réseau en écoute (mode *promiscuous*), il est possible de prendre connaissance de tout le trafic.

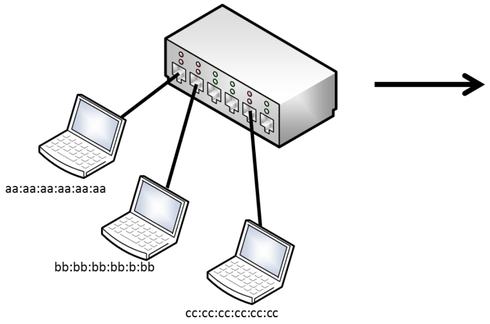
Ethernet et les réseaux commutés

Le mode commuté

Dans ce mode, le support n'est pas partagé. Deux terminaux s'échanges les trames sur une liaison « dédiée ». Le trafic n'est plus systématiquement dupliqué sur tous les ports et il n'y a plus de collision.

Les terminaux sont reliés à un commutateur (un *switch*) qui analyse les trames pour ne les transmettre qu'aux stations cibles, identifiées par leur adresse MAC.

Pour effectuer cette tâche, le commutateur maintient une table de correspondance **adresse MAC de la station / port de connexion** (appelée table CAM ou table des adresses MAC).



Port	@ Mac
1	aa:aa:aa:aa:aa:aa
2	bb:bb:bb:bb:b:bb
3	
4	
5	cc:cc:cc:cc:cc:cc
6	

Ethernet et les réseaux commutés

Le mode commuté

L'utilisation de commutateurs permet d'assurer un premier niveau de segmentation : il n'est en théorie plus possible d'écouter le trafic qui ne nous est pas destiné. En pratique c'est insuffisant :

- Certains commutateurs se transforment en HUB lorsqu'il n'arrivent plus à tenir la charge de trafic (ou parfois parce qu'on ne sait pas pourquoi) ;
- Il existe des attaques ciblant les commutateurs ou les autres hôtes du réseau permettant de récupérer illégitimement le trafic.

L'usage d'un réseau commuté ne permet à lui seul de garantir la sécurité des échanges.

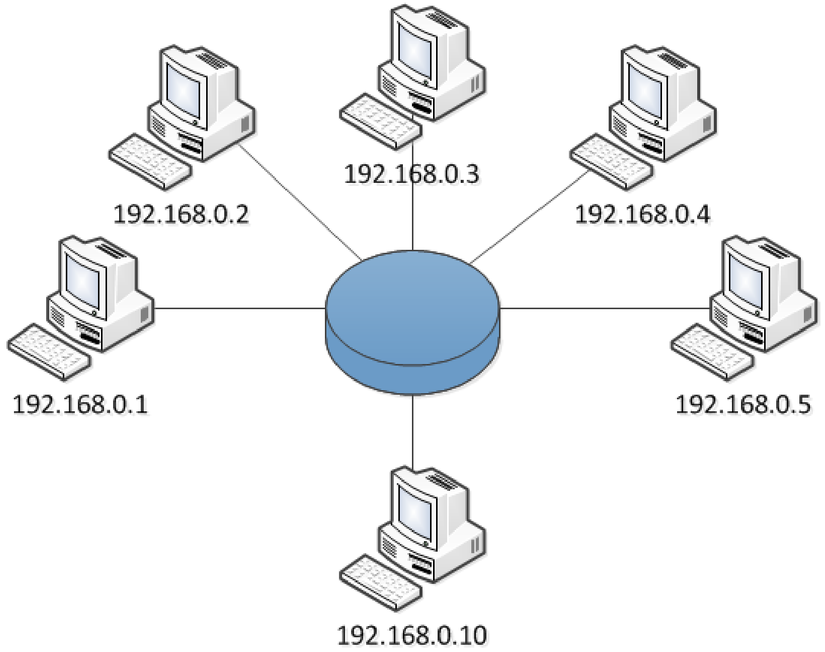
MAC flooding & duplicating

- Le ***MAC flooding*** consiste à émettre un très grand nombre de trames Ethernet avec de fausses adresses MAC. La table de correspondance ports/adresses du switch ayant une taille limitée, il est possible alors de la saturer. Dans ce cas, l'équipement peut soit tomber (déni service), soit se transformer en hub et dupliquer le trafic sur l'ensemble de ces ports.

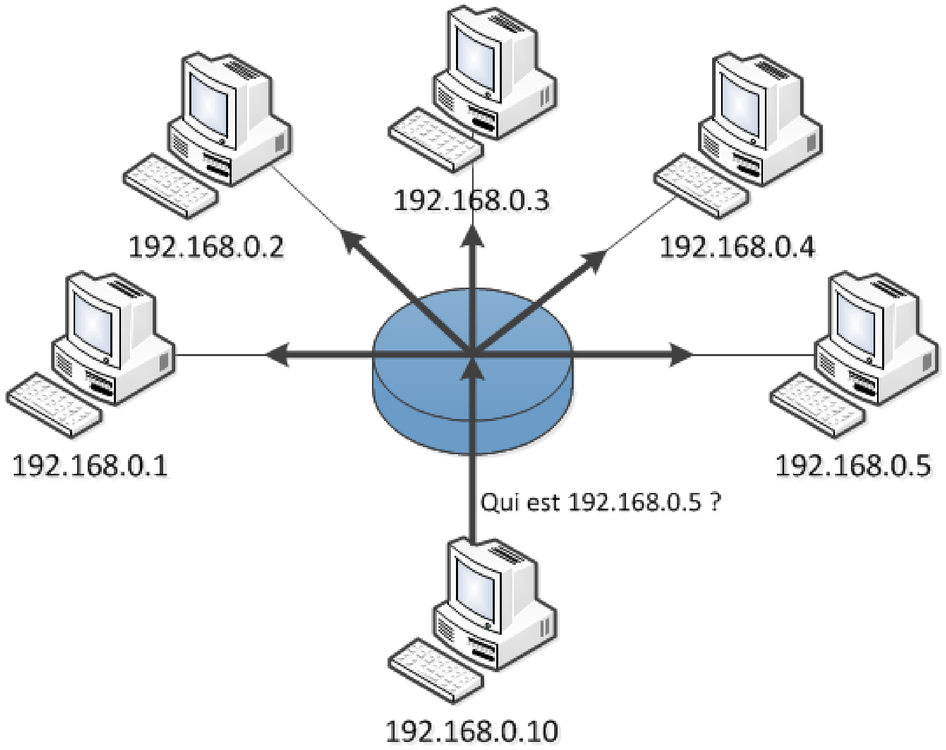
- Le ***MAC duplicating*** (ou *cloning*) consiste à affecter à sa carte réseau l'adresse MAC de la victime. Le switch a alors deux entrées dans sa table de correspondance ports/adresses avec la même MAC. Dans certains cas, il est susceptible de dupliquer le trafic à destination de la victime sur le port de l'attaquant.

Le protocole ARP

L'*Address Resolution Protocol* (ARP) a été défini dans la RFC 826. Il est utilisé pour traduire une adresse du protocole de la couche réseau (IP le plus souvent) vers une adresse de la couche liaison (Ethernet par exemple).



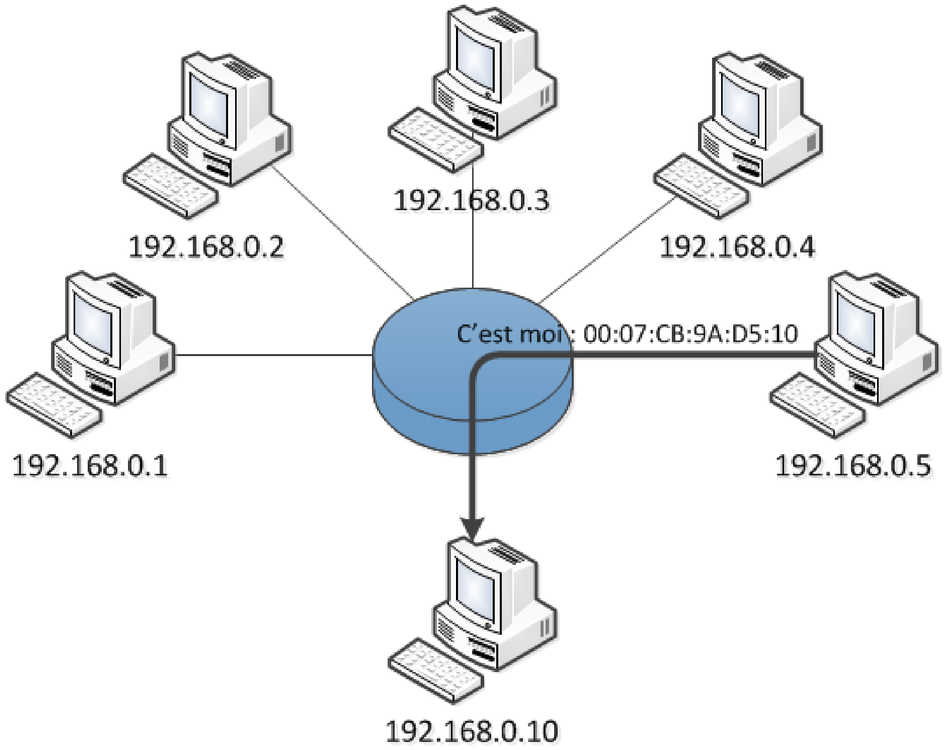
Le protocole ARP



- La machine **192.168.0.10** souhaite échanger avec la machine **192.168.0.5** La couche liaison de données a besoin de connaître son adresse MAC.

- Elle va envoyer une requête ARP en broadcast sur le réseau à destination de 192.168.0.255 et l'ensemble des machines va la recevoir.

Le protocole ARP



- La machine 192.168.0.5 va répondre et donner son adresse MAC (00:07:CB:9A:D5:10)
- La machine 192.168.0.10 va envoyer sa trame à destination de 00:07:CB:9A:D5:10

Le protocole ARP

Sur de nombreux systèmes, ARP embarque un cache, qui conserve le résultat des dernières requêtes. Il s'agit d'une liste de correspondances entre adresses MAC et adresses IP.

Lorsque la pile IP a besoin de traduire une adresse IP en adresse MAC, afin d'émettre une trame, elle va d'abord solliciter le cache. Si l'adresse IP est présente, aucune requête ARP n'est effectuée.

La commande *arp* permet de manipuler ce cache sur le système local :

```
# arp -a
? (192.168.0.5) at 00:07:cb:9a:d5:10 [ether] on eth0
? (192.168.0.2) at 00:06:5b:39:ef:21 [ether] on eth0
```

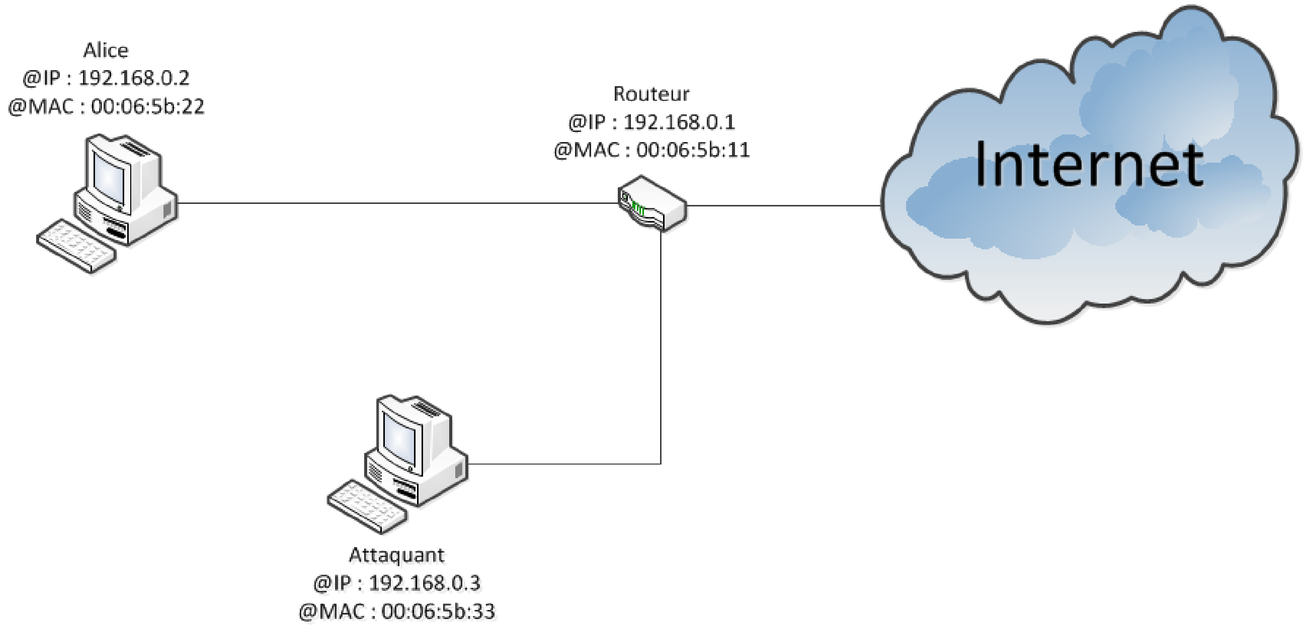
ARP Cache Poisonning

Une attaque de type *ARP Cache Poisonning* vise à empoisonner le cache ARP d'une machine en y déposant des associations adresse IP/adresse MAC fallacieuses. Au moment de traduire une adresse IP en adresse MAC, la pile IP de la victime va inspecter son cache ARP et va utiliser l'adresse qui a été insérée par l'attaquant et donc envoyer sa trame à un mauvais destinataire.

Cela permet de détourner le trafic de la victime vers. Comme l'attaquant usurpe l'adresse MAC d'un hôte du réseau, cette attaque est aussi parfois appelée *ARP Spoofing*.

Cela est possible car ARP est un protocole *stateless* (sans état) : il n'est pas capable d'associer une réponse reçue à une requête précédemment émise. Après l'émission d'une requête en broadcast, il inscrit au cache ARP toutes les réponses qu'il reçoit. Il est donc facile d'abuser de ce protocole en envoyant des réponses non-sollicitées

ARP Cache Poisonning



- Un attaquant souhaite écouter les communications d'une victime (Alice) sur Internet ;
- Il va donc **empoisonner** le cache de la **victime** et de la **passerelle Internet** (pour récupérer le flux de communication dans les deux sens).

ARP Cache Poisonning

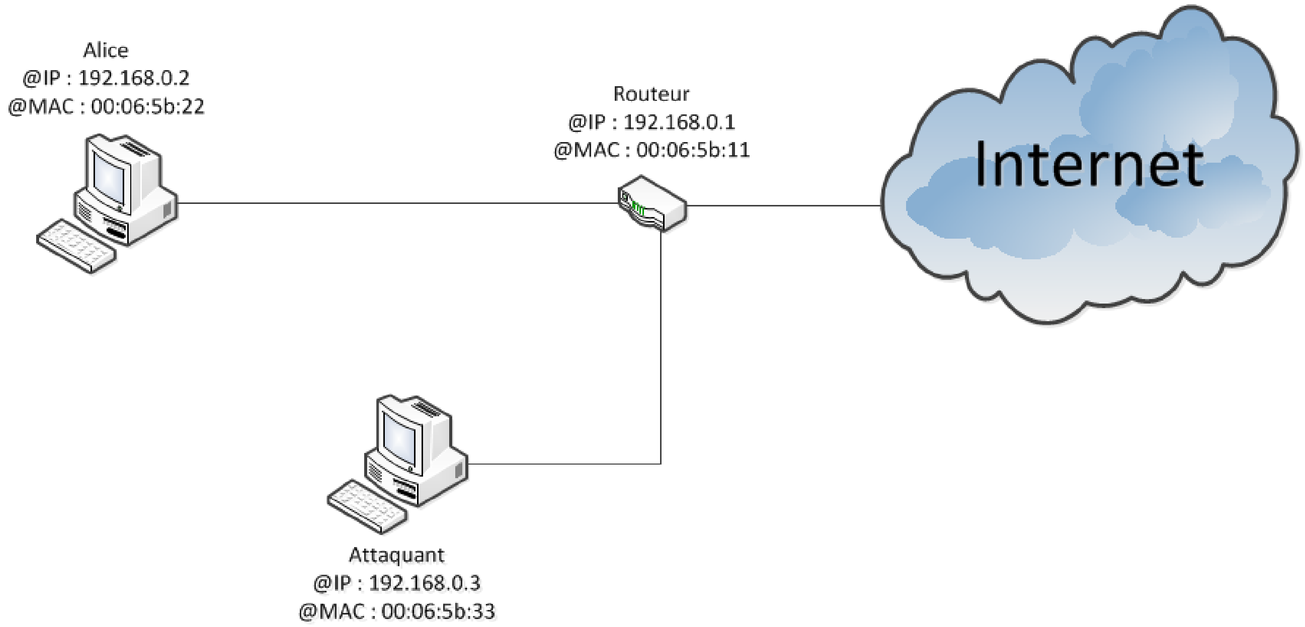


Table ARP d'Alice

? (192.168.0.1) at 00:06:5b:11

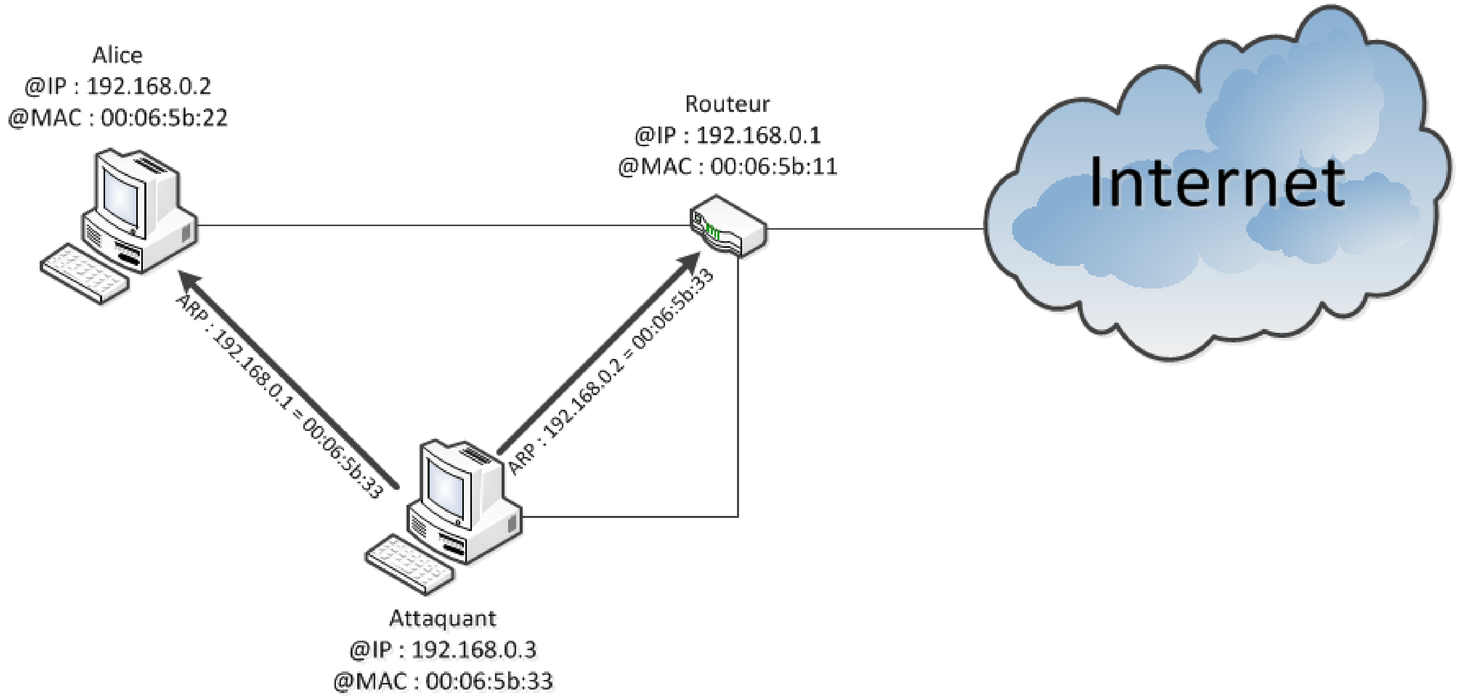
? (192.168.0.3) at 00:06:5b:33

Table ARP de la passerelle

? (192.168.0.3) at 00:06:5b:33

? (192.168.0.2) at 00:06:5b:22

ARP Cache Poisonning



L'attaquant envoie deux réponses ARP non-sollicitées, associant son adresse MAC...

- Avec l'adresse IP de la passerelle auprès d'Alice ;
- Avec l'adresse IP d'Alice auprès de la passerelle.

ARP Cache Poisonning

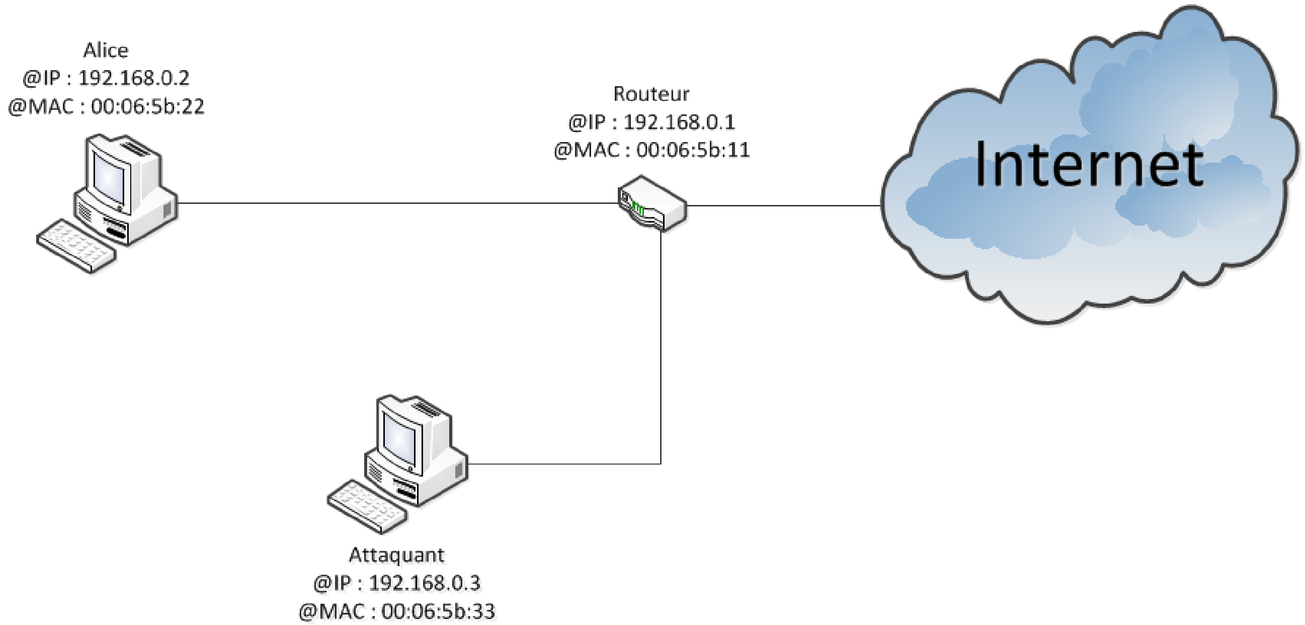


Table ARP d'Alice

? (192.168.0.1) at 00:06:5b:33

? (192.168.0.3) at 00:06:5b:33

Table ARP de la passerelle

? (192.168.0.3) at 00:06:5b:33

? (192.168.0.2) at 00:06:5b:33

ARP Cache Poisonning

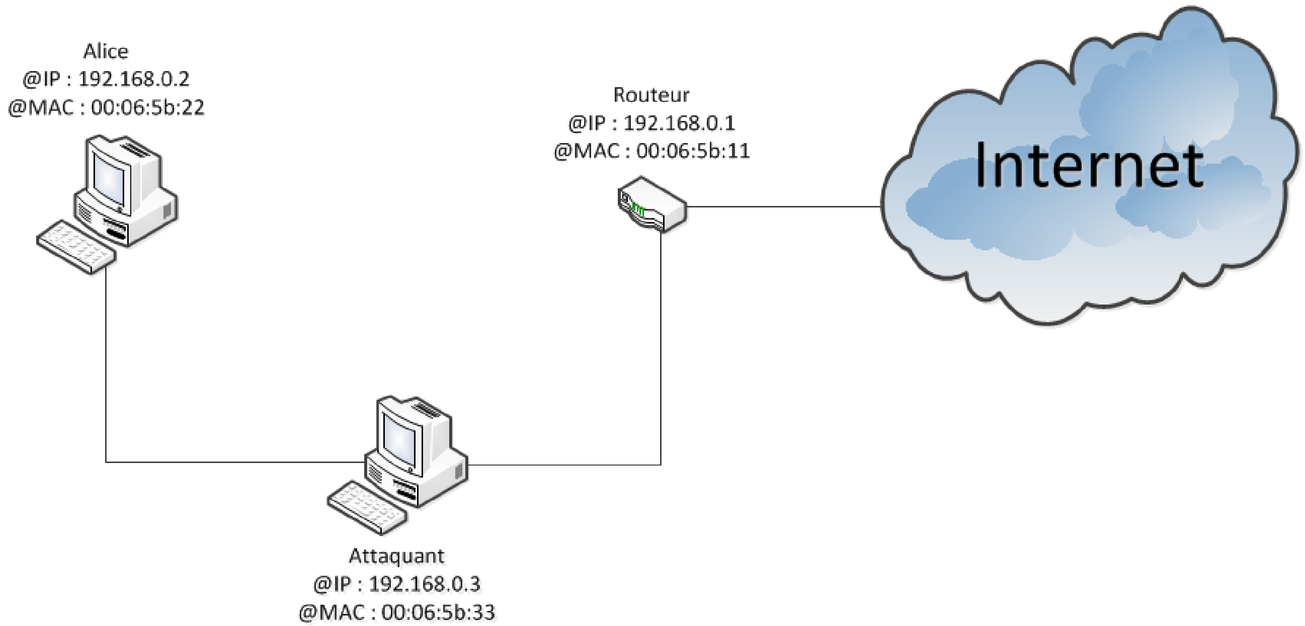


Table ARP d'Alice

? (192.168.0.1) at 00:06:5b:33

? (192.168.0.3) at 00:06:5b:33

Table ARP de la passerelle

? (192.168.0.3) at 00:06:5b:33

? (192.168.0.2) at 00:06:5b:33

ARP Cache Poisonning

L'empoisonnement du cache a placé l'attaquant en coupure de flux entre Alice et la passerelle :

- Alice va envoyer ses trames destinées à Internet à l'adresse IP de la passerelle par défaut définie dans sa configuration. Suite à l'empoisonnement et après la résolution de l'adresse IP en adresse MAC, Alice va envoyer ces trames à l'attaquant ;

- L'attaquant est en mesure de prendre connaissance des messages envoyés par Alice, de les bloquer (déni de service), de les modifier si besoin et de les réémettre vers la bonne passerelle (puisqu'il connaît la bonne adresse MAC) ;

- De même, les réponses de la passerelle seront transférées à l'attaquant, pour les mêmes raisons, qui pourra les faires suivre à Alice.

L'attaquant est en position « d'homme du milieu » (en anglais *Man in the Middle*).

ARP Cache Poisonning

Comment se protéger ?

- En inscrivant les adresses MAC des hôtes de façon statique dans les tables ARP des machines du réseau (peu pratique)
- En utilisant des équipements réseaux professionnels qui embarquent des protections :
  - **Port security**, permet de limiter le nombre d'adresses MAC connectées à une même interface d'accès ;
  - **DHCP snooping**, qui consiste à déclarer des ports de confiance identifiés comme les seuls par lesquels peuvent provenir des baux DHCP, maintenir une table d'association afin de conserver l'état des baux attribués aux terminaux et limiter le nombre de requêtes DHCP par seconde sur une interface ;
  - **IP Source Guard**, consistant à vérifier la cohérence entre les adresses IP utilisées par les terminaux connectés au commutateur et les données contenues dans la table DHCP snooping, afin d'empêcher l'IP spoofing ;
  - **ARP inspection**, qui consiste à vérifier la cohérence entre le contenu des trames ARP (adresses source MAC/IP et adresses destination MAC/IP) et les données contenues dans la table DHCP snooping, afin de détecter et bloquer les tentatives d'ARP spoofing.

Rappel sur ICMP

Le protocole ICMP (*Internet Control Message Protocol*) est utilisé pour véhiculer des messages de contrôle et d'erreur pour le compte de TCP/IP. Bien qu'il se situe dans la pratique au-dessus d'IP, il s'agit tout de même d'un protocole de niveau 3.

Les messages transportés sont de différentes natures. On distingue 14 différents types. Les plus connus sont :

- Type 8 – echo (*ping*) ;
- Type 0 – réponse echo (*pong*) ;
- Type 3 – destinataire inaccessible.

Le **type 5** est particulièrement intéressant, car il permet d'envoyer une demande de modification dynamique de la configuration de routage de la cible. Il est normalement utilisé lorsqu'une incohérence est détectée par un routeur, pour permettre à un client du réseau d'utiliser une route plus courte.

Rappel sur ICMP

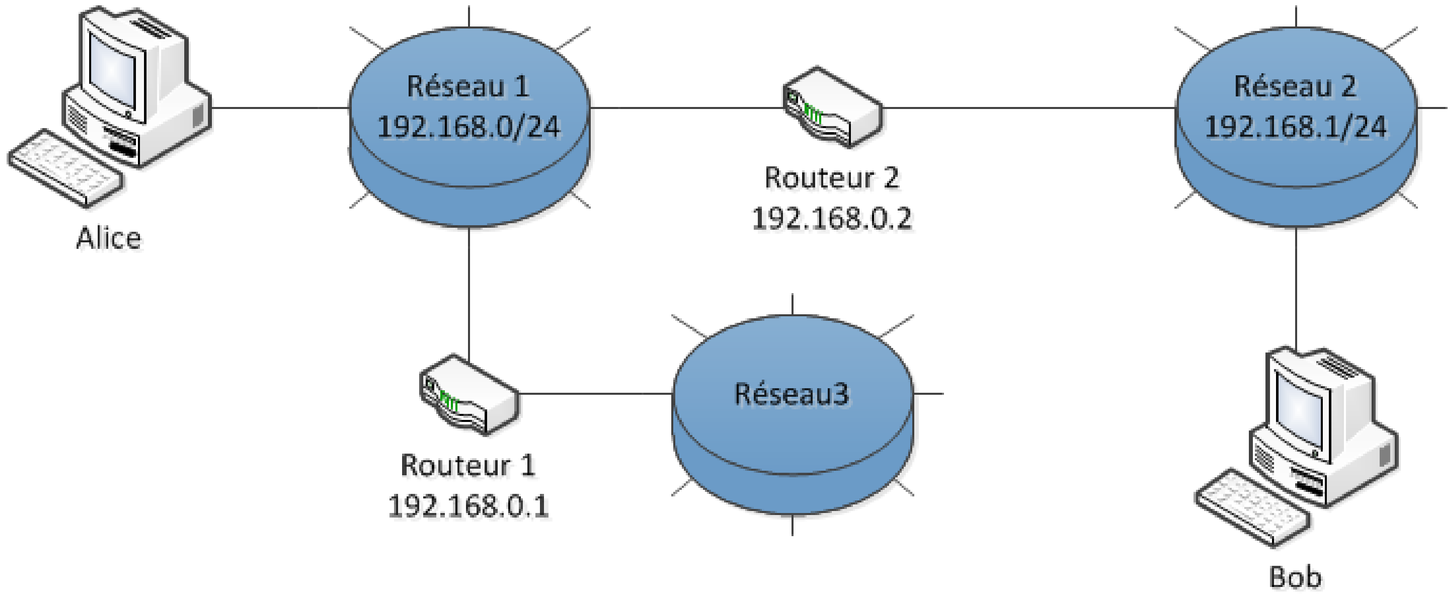


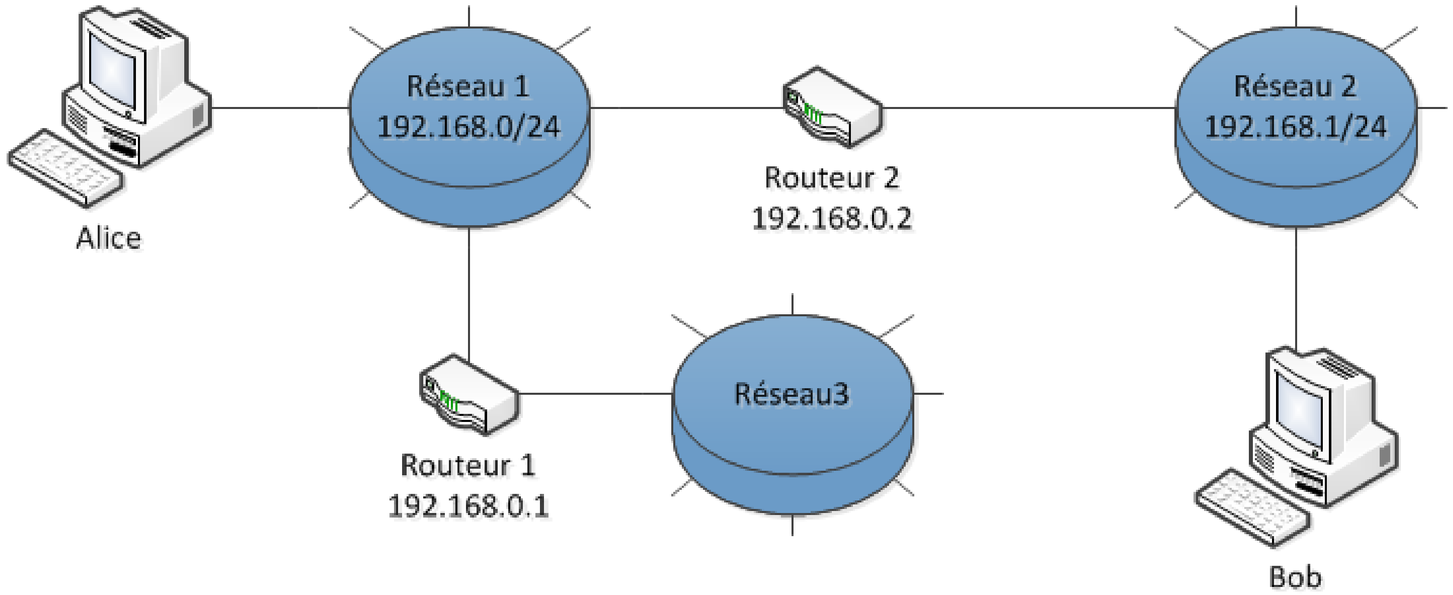
Table de routage d'Alice :

Destination	Passerelle	Genmask
192.168.1.0	192.168.0.1	255.255.255.0

Table de routage du routeur 1 :

Destination	Passerelle	Genmask
192.168.1.0	192.168.0.2	255.255.255.0

Rappel sur ICMP



Si Alice utilise le routeur 1 pour s'adresser à Bob, il détectera que le prochain routeur (le routeur 2) appartient également au réseau 1. Il enverra un *ICMP Redirect* à Alice, pour qu'elle passe directement par le routeur 2

Rappel sur ICMP

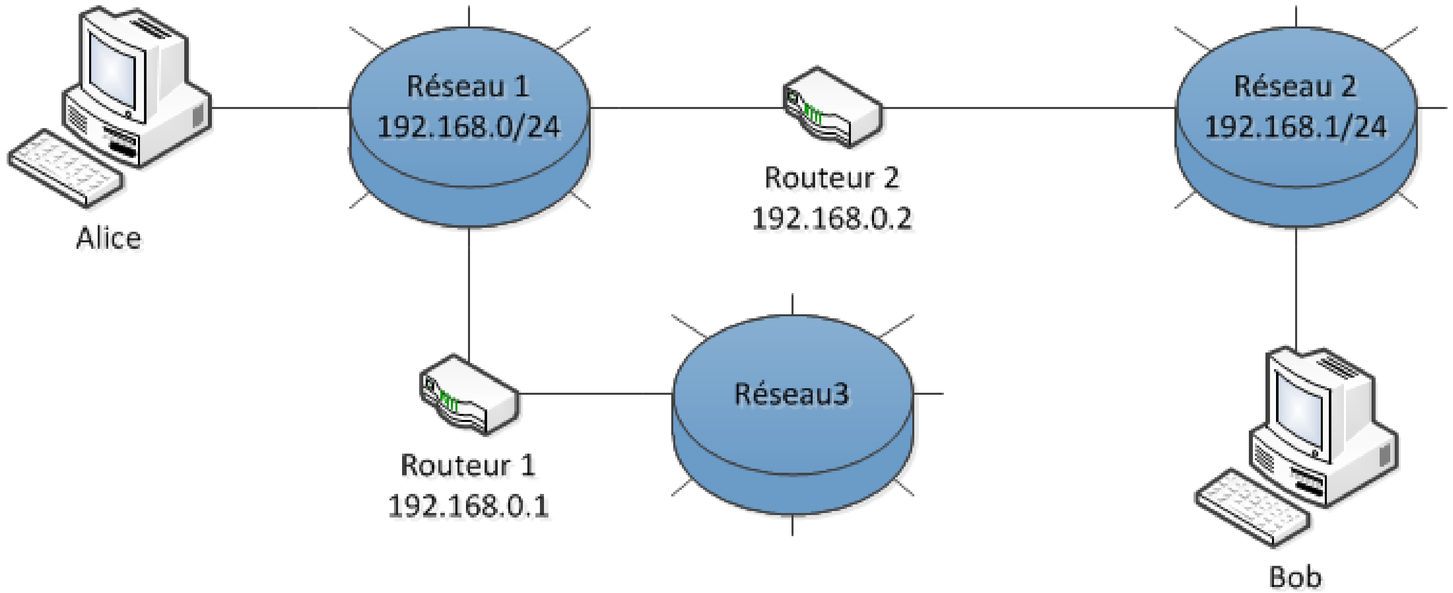


Table de routage d'Alice :

Destination	passerelle	Genmask
192.168.1.0	192.168.0.2	255.255.255.0

Alice a fait évoluer sa table de routage

ICMP Redirect

Une attaque de type *ICMP Redirect* vise à abuser de cette fonctionnalité légitime pour se placer en coupure de flux de deux autres machines. Il suffit d'envoyer un message de **type 5** à chaque victime pour qu'elle modifie sa table de routage.

Pour être transparent dans la réalisation du *MitM*, les datagrammes à destination de l'autre victime doivent être routés par notre machine. Pour cela, il faut configurer sa machine pour relayer les flux

```
# sysctl -w net.ipv4.ip_forward=1
```

ICMP Redirect

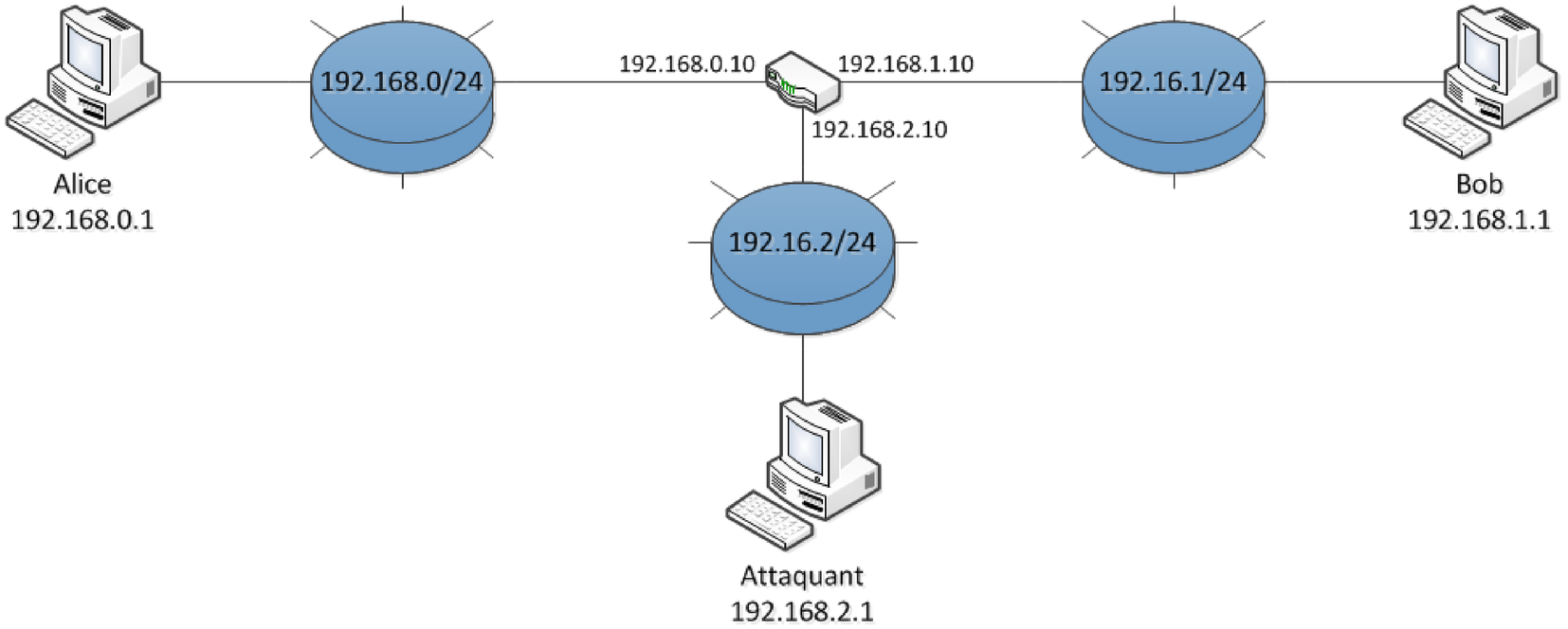


Table de routage d'Alice

Destination Passerelle Genmask

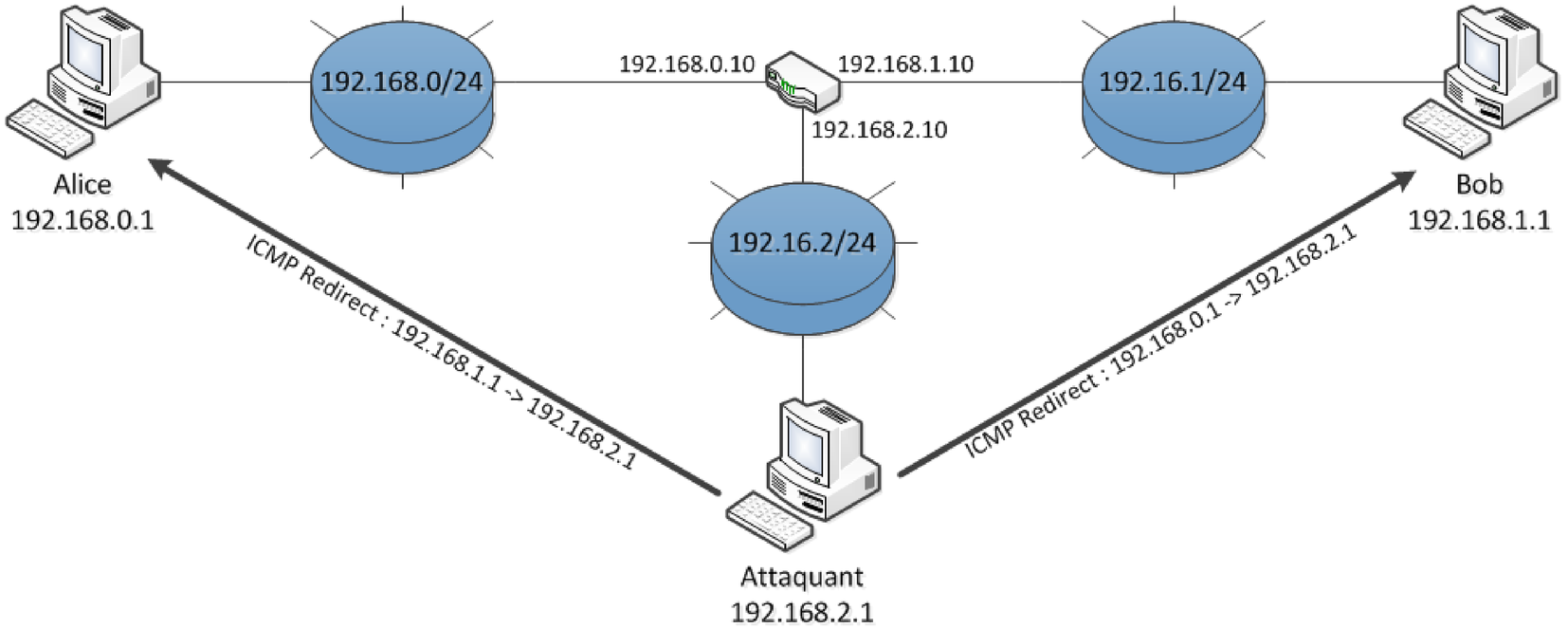
192.168.1.0 192.168.0.10 255.255.255.0

Table de routage de Bob

Destination Passerelle Genmask

192.168.0.0 192.168.1.10 255.255.255.0

ICMP Redirect



L'attaquant lance une attaque sur Alice et Bob en leur envoyant des ***ICMP Redirect*** **illégitimes** :

- Alice : il faut passer par 192.168.2.1 pour atteindre Bob
- Bob : il faut passer par 192.168.2.1 pour atteindre Alice

ICMP Redirect

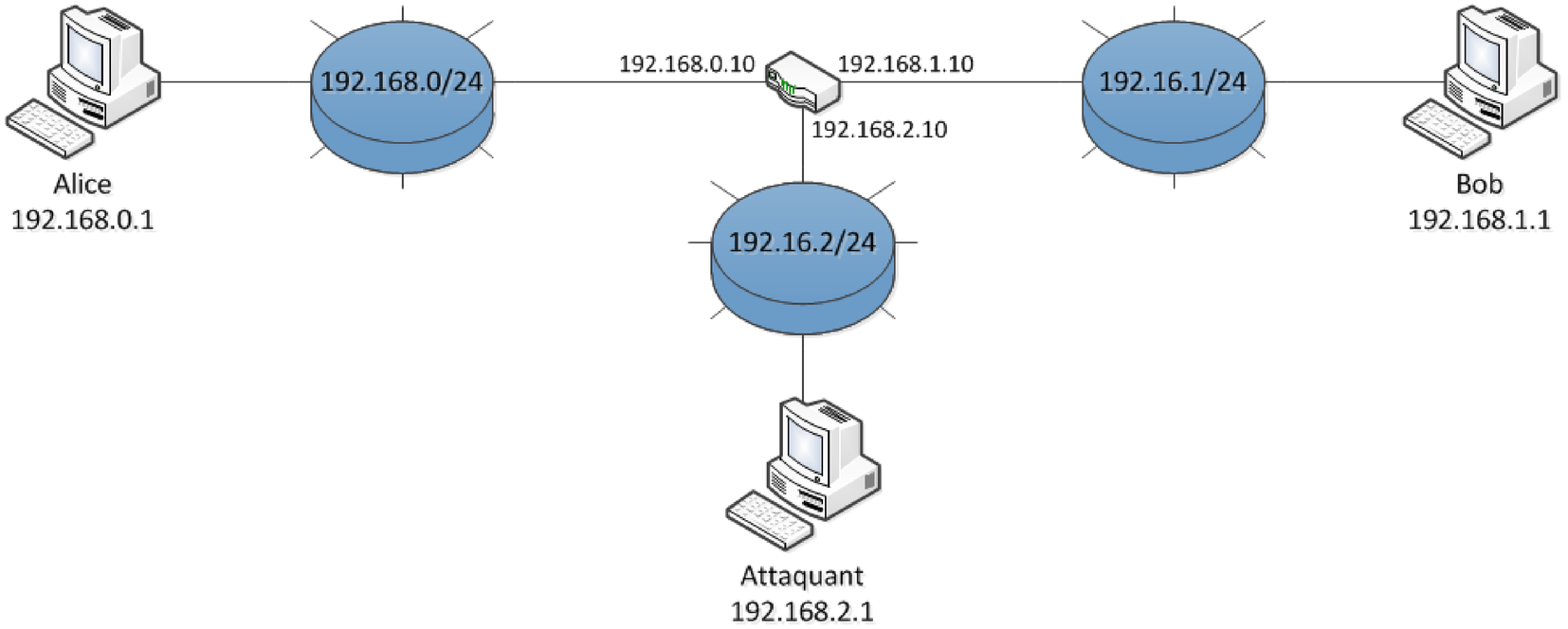


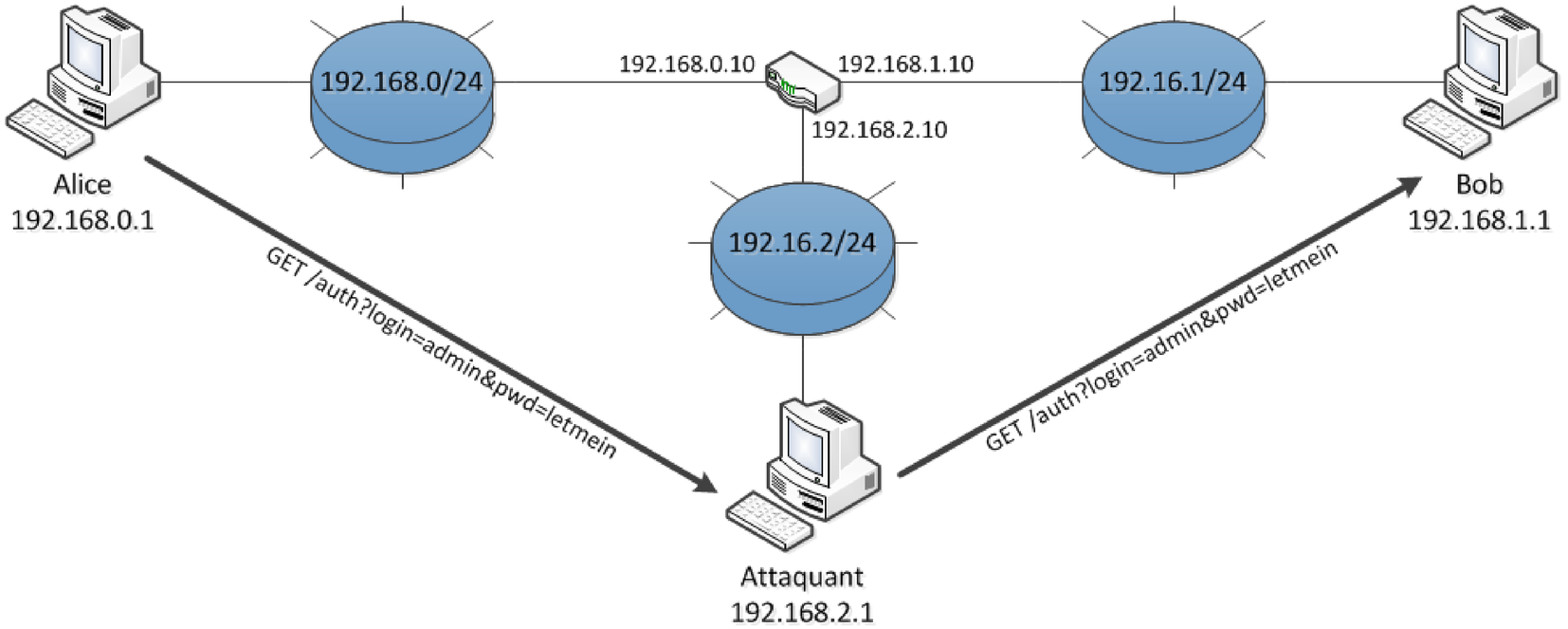
Table de routage d'Alice

Destination	Passerelle	Genmask
192.168.1.0	192.168.0.10	255.255.255.0
192.168.1.1	192.168.2.1	255.255.255.255

Table de routage de Bob

Destination	Passerelle	Genmask
192.168.0.0	192.168.1.10	255.255.255.0
192.168.0.1	192.168.2.1	255.255.255.255

ICMP Redirect



Désormais, lorsqu'Alice envoie un paquet à Bob (et inversement), il **transite par la machine de l'attaquant**, qui relaie les flux :

- L'attaque est **transparente** ;
- L'attaquant peut **capturer**, voire **modifier**, l'ensemble des flux.

ICMP Redirect

Comment se protéger ?

Il existe des options permettant de ne pas prendre en compte les ICMP Redirect illégitimes :

- **net.ipv4.conf.all.accept_redirects**

  Positionnée à 0, cette option désactivera la prise en compte des ICMP Redirect sur la machine

- **net.ipv4.conf.all.secure_redirects**

  Positionnée à 1, cette option ne prendra en compte que les ICMP Redirect provenant de machines identifiées comme des passerelles

- **EnableICMPRedirect**

  Sous Windows, la prise en charge des messages ICMP Redirect est également faite par défaut. Il faut positionner cette option à 0.

Outillage

En terme d'outillage, les utilitaires simples, gratuits et efficaces ne manquent pas. Ils peuvent être de plusieurs types

- Outils de capture de trafic ;
- Outils de manipulation/création de trames ;
- Outils d'attaque ;
- Etc.

Ils existent sous Windows et sous Linux et sont relativement simple d’utilisation.

tcpdump

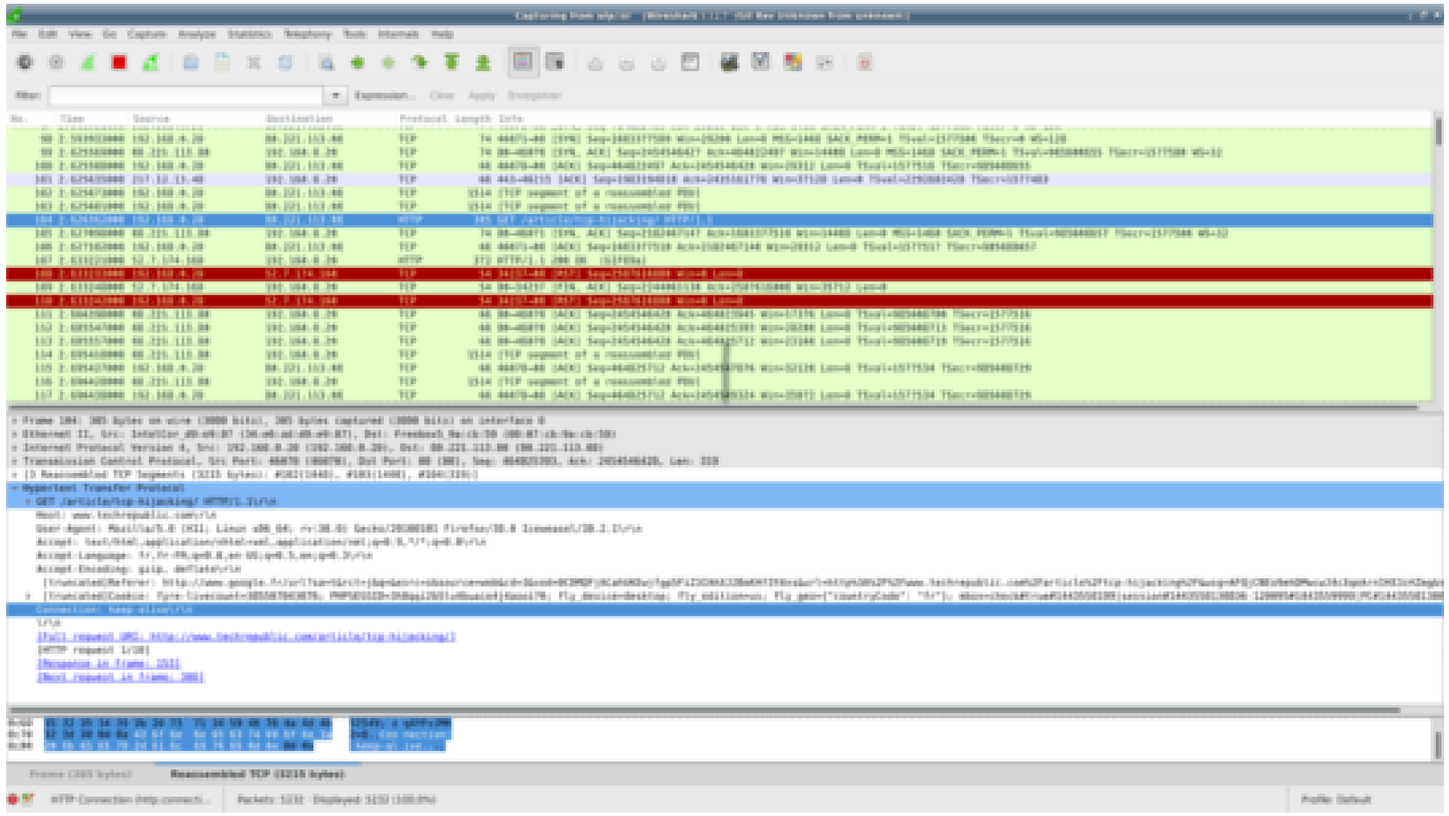
Outil en ligne de commande, simple et complet.

Attention au piège classique de l'utilisation en SSH.

```
02:18:58.927910 ARP, Request who-has laptop-yves tell gateway, length 28
02:18:58.927922 ARP, Reply laptop-yves is-at 34:e6:ad:d9:e9:87 (oui Unknown), le
02:18:58.928004 IP laptop-yves.48572 > google-public-dns-a.google.com.domain: 49
02:18:58.956225 IP google-public-dns-a.google.com.domain > laptop-yves.48572: 49
02:19:14.477022 IP laptop-yves.38065 > google-public-dns-a.google.com.domain: 33
02:19:14.477039 IP laptop-yves.38065 > google-public-dns-a.google.com.domain: 38
02:19:14.520769 IP google-public-dns-a.google.com.domain > laptop-yves.38065: 33
02:19:14.520787 IP google-public-dns-a.google.com.domain > laptop-yves.38065: 38
02:19:14.563940 IP laptop-yves.55450 > google-public-dns-a.google.com.domain: 68
02:19:14.563980 IP laptop-yves.55450 > google-public-dns-a.google.com.domain: 53
02:19:14.594057 IP google-public-dns-a.google.com.domain > laptop-yves.55450: 68
02:19:14.599001 IP google-public-dns-a.google.com.domain > laptop-yves.55450: 53
02:19:14.599839 IP laptop-yves.41538 > par10s22-in-f228.1e100.net.https: Flags [
02:19:14.600094 IP laptop-yves.50641 > google-public-dns-a.google.com.domain: 42
```

Wireshark (anciennement ethereal)

Outil d'analyse de trafic très complet, avec interface graphique



hping3

- Outil de création et manipulation de trafic, avec des options pour TCP/UDP, IP et ICMP ;
- Plutôt destiné à mener des attaques ;
- Très complet, propose un grand nombre d'options pratiques (flood, spoof, flags TCP, numéros de séquence, d'acquittement, etc.) ;
- Il est très rapide de lancer des attaques classiques avec des one-liners :

```
# hping3 192.168.0.10 -a 192.168.0.25 -p 139 -S --flood
```

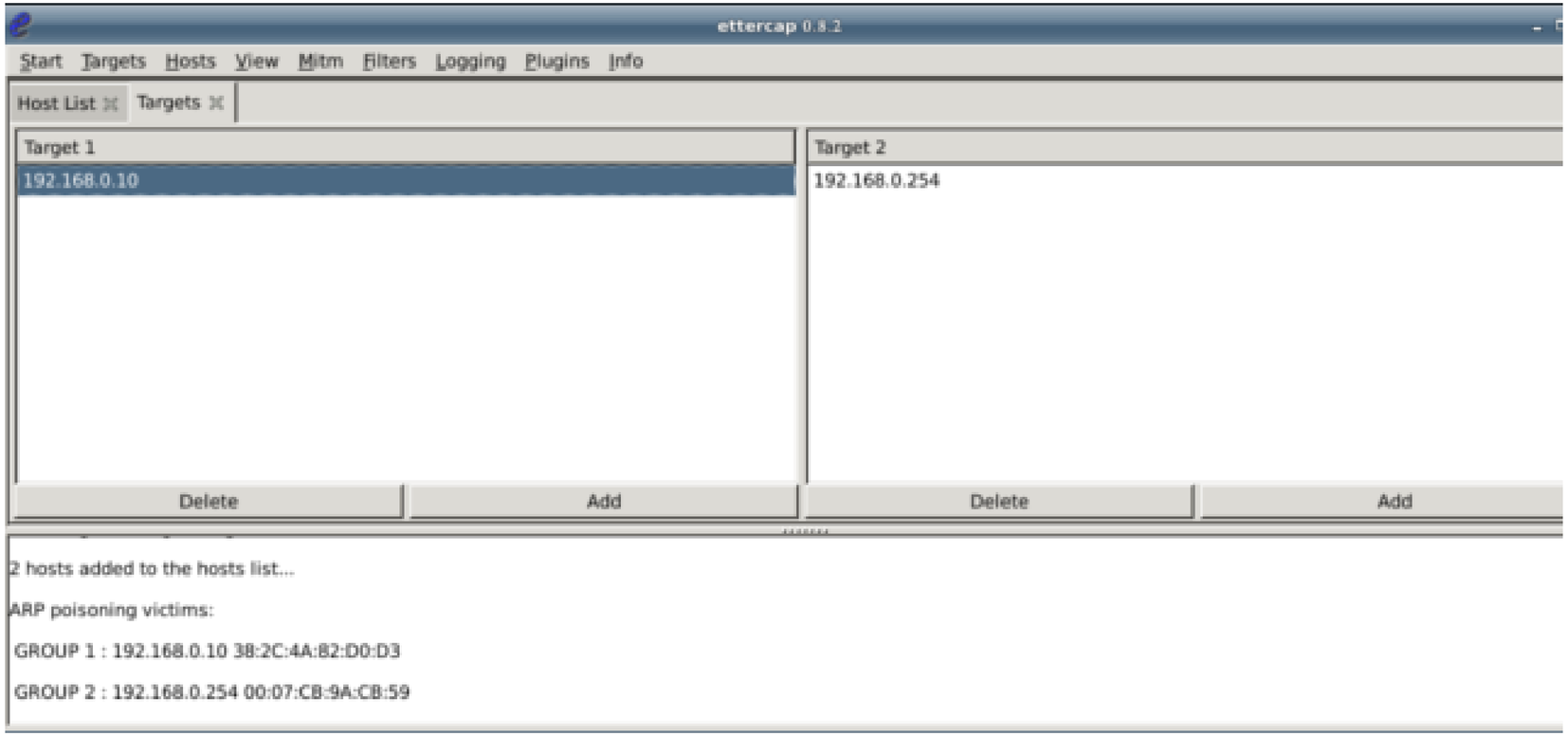
scapy

Librairie de génération et d'analyse de flux en python, avec interface ligne de commande.

```
yves@laptop-yves:~$ sudo scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
INFO: Can't import python Crypto lib. Won't be able to decrypt WEP.
INFO: Can't import python Crypto lib. Disabled certificate manipulation tools
Welcome to Scapy (2.2.0)
>>> p = Ether()/IP(src="192.168.0.25", dst="192.168.0.10")/TCP(flags="S", dport=
>>> for i in range(1, 20):
...   sendp(p)
...
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
```

Ettercap

Outil d'attaques Man In The Middle, avec interface graphique (ARP Poisoning, ICMP Redirect, etc.).



acceis

Merci de votre attention.

Gwenn Feunteun

gwenn@acceis.fr