UNIVERSITÉ CAEN NORMANDIE

NAT & PAT 30.01.2023

Ou l'art de la dissimulation ;)

On avance masqué ☹

Auteur : Pascal Fougeray

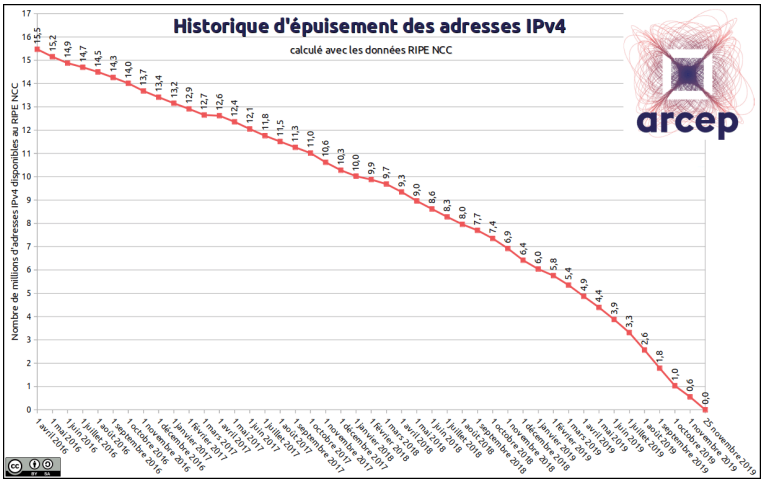
source : https://la-cuisine-de-nat-et-pat.webnode.fr/

1 Introduction

Depuis 2011, il n'y a plus de /22 de disponible pour les LIR.
Plus d'@IPv4 de disponible à la vente ☺
Et... nous ne sommes toujours pas en IPv6.
Il y a quelque chose que je ne comprends pas, si on compte toutes les machines connectées à Internet

— Les serveurs
— Les clients des particuliers et des professionnels
— Les smartphones

Et bien il y a longtemps que nous avons dépassé le nombre d' @IPv4 publiques disponibles ?
https://www.arcep.fr/la-regulation/grands-dossiers-internet-et-numerique/lipv6/suivi-de-lepuisement-des-adresses-ipv4.html



source :
https://www.arcep.fr/fileadmin/cru-1671101953/user_upload/grands_dossiers/ipv6/date_epuisement_ipv4.png

Si vous voulez comprendre le NAT voici une petite vidéo de 16 mn ☺
https://www.youtube.com/watch?v=XFStyE3cafE

2 Introduction

Le NAT (***Network Adress Translation***) résout 2 problèmes, le second problème étant généré par le premier.

1. La pénurie d'adresses IP (en Ipv4 !) sur Internet auquel une réponse apportée a été de spécialiser certaines plages d'adresses IP pour une utilisation privée.
2. Cela a généré le second problème qui est de pouvoir accéder à Internet en utilisant ces adresses IP privées qui **rappelons le ne sont pas routables.** (Bon on n'a pas encore fait le routage mais on fait comme si ☺)

Le ***port forwarding*** consiste à rediriger un port d'une machine vers un port donné sur une machine locale ayant généralement une adresse IP privée.

3 Le NAT

On l'appelle aussi le **camouflage IP** ou ***IP Masquerading***

Le camouflage réécrit les paquets IP lorsqu'ils traversent la passerelle.

Ils semblent toujours provenir de la passerelle elle-même. Le camouflage réécrit ensuite les réponses afin qu'elles semblent venir du destinataire originel.

NAT et IPv6

L'IETF n'encourage pas à utiliser le NAT avec IPv6 du fait que l'espace d'adresse IPv6 est tel que l'économie d'adresse n'est plus une nécessité !

Alors pourquoi nous ne sommes pas tous en IPv6 [1] ?...

3.1 La solution temporaire

L'organisme qui gère Internet (**IANA**) a décidé que les machines qui ne sont pas serveurs n'ont pas besoin d'avoir d'adresses IP publiques vu qu'elles ne délivrent aucun service. Elles ne sont pas des serveurs, juste des clients !

On va donc utiliser l'adressage privé

Bloc	Usage	Total
10.0.0.0/8	**Adresses privées**	16 777 216
172.16.0.0/12	**Adresses privées**	1048576
192.168.0.0/16	**Adresses privées**	65536

...

Pour rappel à l'Université, du moins au département informatique vos machines sont sur le réseau 10.38.16.0/22 qui est un réseau privé puisqu'il est inclus dans le réseau 10.0.0.0/8.

Il permet la mise en place de 1024 -2 (Rx et ***Broadcast***) = 1022 machines différentes.

...

3.2 Le fonctionnement

Il est fort simple à comprendre !

La machine, souvent un routeur, qui sera reliée à Internet, possède 2 réseaux :

1. Un **privé** en interne, qui n'est pas accessible de l'extérieur !
2. Un **public** second en externe, donc accessible de l'extérieur

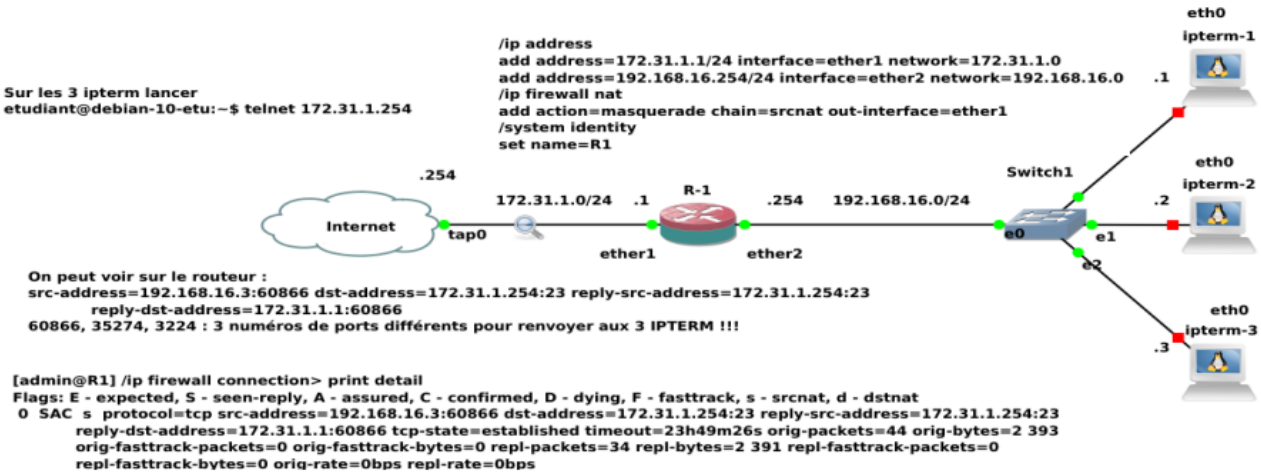
Ce routeur va masquer l'adresse IP privée des clients (les machines à l'intérieur !) et la remplacer par la sienne !

On parle de translation d'adresse

Prenons l'exemple suivant : Un TP que vous pouvez faire seul dans la VM

3 clients (ici des IPTerm) décident de faire du ***Telnet*** (avec du **SSH** cela fonctionne aussi !)

1. Je dis ça je n'ai rien dit ☺



Voici ce qu'il en résulte
La commande ***/ip firewall connection print*** sur le routeur renvoie :

```
[admin@R1] > /ip firewall connection print
Flags: E - expected, S - seen-reply, A - assured, C - confirmed,
D - dying, F - fasttrack, s - srcnat, d - dstnat
 #          PR.. SRC-ADDRESS           DST-ADDRESS           TCP-STATE
 0  SAC  s  tcp  192.168.16.3:42104    172.31.1.254:23       established
 1  SAC     tcp  172.31.1.254:34278    172.31.1.1:8291       established
 2  SAC  s  tcp  192.168.16.1:45560    172.31.1.254:23       established
 3  SAC  s  tcp  192.168.16.2:40424    172.31.1.254:23       established
```

On constate 3 connexions sur le port 23 du serveur Telnet, ici l'interface **Tap0** de la VM qui a pour IP **172.31.1.254**

Remarque : La connexion sur le port 8291 correspond à la connexion winbox et n'a rien à voir !

Sur le serveur ***Telnetd*** donc la VM, la commande ***netstat -ltpan4 | grep 23*** renvoie :
On constate les 3 mêmes numéros de port !
On peut en conclure qu'il n'y a qu'une translation d'adresses !
Une capture wireshark entre le routeur et l'interface Tap0 de la VM qui représente Internet montre les communications et les 3 ports

```
… 253.067650  172.31.1.254  172.31.1.1      TELNET    138 Telnet Data ...
… 253.068772  172.31.1.1    172.31.1.254    TCP        66 45560 → 23 [ACK] Seq=20 Ack=1092 Win=870 Len=0 TSval=3190761395 TSecr=2231658831
… 255.218828  172.31.1.1    172.31.1.254    TELNET     67 Telnet Data ...
… 255.219324  172.31.1.254  172.31.1.1      TELNET     67 Telnet Data ...
… 255.220747  172.31.1.1    172.31.1.254    TCP        66 40424 → 23 [ACK] Seq=18 Ack=940 Win=501 Len=0 TSval=605010584 TSecr=2231660983
… 255.359405  172.31.1.1    172.31.1.254    TELNET     67 Telnet Data ...
… 257.910295  172.31.1.1    172.31.1.254    TELNET     67 Telnet Data ...
… 257.911223  172.31.1.254  172.31.1.1      TELNET     67 Telnet Data ...
… 257.913121  172.31.1.1    172.31.1.254    TCP        66 42104 → 23 [ACK] Seq=19 Ack=1019 Win=501 Len=0 TSval=3161455110 TSecr=2231663674
… 258.061547  172.31.1.1    172.31.1.254    TELNET     67 Telnet Data ...
… 258.062169  172.31.1.254  172.31.1.1      TELNET     67 Telnet Data ...
… 255.810211  172.31.1.254  172.31.1.1      TELNET    138 Telnet Data ...
… 255.811701  172.31.1.1    172.31.1.254    TCP        66 40424 → 23 [ACK] Seq=21 Ack=1015 Win=501 Len=0 TSval=605011174 TSecr=2231661574
```

3.3 La configuration

Cela dépend du système que l'on utilise

3.3.1 Avec Mikrotik

Sur un routeur tel celui utilisé en TP, ce n'est qu'une seule ligne :
/ip firewall nat add action=masquerade chain=srcnat out-interface=ether1

Il est facile de comprendre que l'on va masquer (**masquerade**) les IP **sources** qui sortent sur l'interface **ether1** en faisant du **nat**

Dans notre cas si on lance sur le routeur qui fait le NAT la commande
/ip firewall connection print detail
on récupère :

```
[admin@R1] /ip firewall> connection print detail
Flags: E - expected, S - seen-reply, A - assured, C - confirmed,
D - dying, F - fasttrack, s - srcnat, d - dstnat
 0  SAC  s  protocol=tcp src-address=192.168.16.3:42104
             dst-address=172.31.1.254:23
             reply-src-address=172.31.1.254:23
             reply-dst-address=172.31.1.1:42104 tcp-state=established
             timeout=23h27m38s orig-packets=126 orig-bytes=6 744
             orig-fasttrack-packets=0 orig-fasttrack-bytes=0
             repl-packets=85 repl-bytes=6 501 repl-fasttrack-packets=0
             repl-fasttrack-bytes=0 orig-rate=0bps repl-rate=0bps
```

Port !!!

Je ne montre que pour le port 42104, la commande renvoie la même chose pour les 2 autres ports !

3.3.2 Avec Linux

Linux possède la capacité de faire du NAT si votre machine fait du routage donc si elle a au moins 2 interfaces réseaux mais pas obligatoirement si on utilise les sous interfaces vues au cours sur les Switchs et les Vlans... !
Il suffit d'utiliser les commandes **iptables**

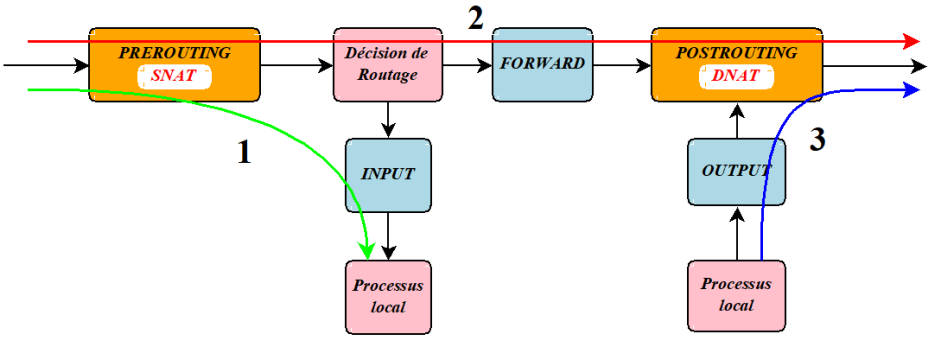
Par défaut la table FILTER est vide et accepte tout.
Aucune règle de translation d'adresse n'est présente par défaut.
Le noyau dispose de listes de règles appelées des chaînes.
Les règles sont analysées les unes à la suite des autres dans l'ordre de leur écriture.
Dès qu'une règle peut s'appliquer à un paquet, elle est déclenchée, et la suite de la chaîne est ignorée.

Les chaînes sont regroupées dans 3 tables.

1. Table NAT (Network Address Translation) : elle est utilisée pour la **translation d'adresse** ou la **translation de port**.
   Deux types de chaînes s'appliquent à cette table :
   (a) **PREROUTING**
   (b) **POSTROUTING**
2. Table **FILTER** : c'est la table par défaut. Elle contient toutes les règles de filtrage.
   Trois types de chaines s'appliquent à cette table :
   (a) **INPUT**
   (b) **OUTPUT**
   (c) **FORWARD**

Voici l'architecture du Firewall Linux (un dessin c'est mieux ^^)

1. Un paquet arrive, on a du "***prérouting***" s'il est pour la machine, la décision de routage **l'enverra ou pa**s au processus local
2. Un paquet arrive, on a du "***prérouting***" s'il est pour une autre machine, la décision de routage le **transfère** en sortie via le "***postrouting***" qui **l'enverra ou pas** vers la sortie
3. Un paquet d'un processus local, doit être envoyé à une autre machine, le "***postrouting***" **l'enverra ou pas** vers la sortie



Un paquet

— rentrera toujours dans la machine via la chaine **PREROUTING**

et

— sortira toujours de la machine via la chaine **POSTROUTING**.

Si le paquet doit être **routé**, il passera dans la chaine **FORWARD**.

Les chaînes **INPUT** et **OUTPUT** quant à elles serviront respectivement à placer des règles pour les paquets **destinés au et émis par le firewall lui-même.**

Pour ajouter une chaîne, il faut avoir pris une décision politique !

On interprète comme cela :

— **-A** ajouter après les autres règles déjà mises

— -o

— -s

— **-j** action (DROP, ACCEPT ou MASQUERADE)

3.3.2.1 Le SNAT ou NAT Source

Les machines internes à l'entreprise ou l'administration disposent

Il substitue une adresse source dans un paquet sortant à son adresse source d'origine.

Dans le cas suivant, on substitue aux requêtes provenant du réseau ***172.16.0.0/24***, une des 10 adresses publiques.

iptables -F INPUT ; iptables -P INPUT ACCEPT
iptables -F OUTPUT ; iptables -P OUTPUT ACCEPT
iptables -F FORWARD ; iptables -P FORWARD ACCEPT
iptables -t nat -F POSTROUTING
iptables -t nat -A POSTROUTING -s 172.16.0.0/24 -j SNAT --to-source 195.115.90.1-195.115.90.10

3.3.2.2 Le DNAT ou NAT Destination

Il substitue à l'adresse de destination des paquets provenant du réseau public, une adresse du réseau local privé.

Dans l'exemple suivant, les paquets à destination de la machine ***193.55.16.64*** sont redirigés vers la machine ***172.16.0.1*** **Le port n'est pas pris en compte**

iptables -F INPUT ; iptables -P INPUT ACCEPT
iptables -F OUTPUT ; iptables -P OUTPUT ACCEPT
iptables -F FORWARD ; iptables -P FORWARD ACCEPT
iptables -t nat -F PREROUTING
iptables -t nat -A PREROUTING -d 193.55.16.64/32 -j DNAT --to-destination 172.16.0.1/32

3.3.2.3 Quelques exemples :

Pour faire quoi	La commande
Interdire en entrée toute requête HTTP	***iptables -A INPUT -s 192.168.0.0/24 -p tcp --dport 80 -j REJECT***
Autoriser en entrée le port 22 SSH sur l'interface eth0	***iptables -A INPUT -p tcp -i eth0 --dport ssh -j ACCEPT***
Autoriser des connexions sortantes HTTP et HTTPS pour les ports TCP 1024 à 65535	***iptables -A OUTPUT -o eth0 -p tcp --dport 80 --sport 1024 :65535 -j ACCEPT***
	iptables -A OUTPUT -o eth0 -p tcp --dport 443 --sport 1024 :65535 -j ACCEPT
Bloquer le réseau 46.161.9.0/24	***iptables -A INPUT -s 46.161.9.0/24 -j DROP***

Remarque :

Les règles créées avec ***iptables*** sont éphémères et ne valent que jusqu'à ce que votre ordinateur est allumé.

On peut utiliser ***iptables-save*** pour sauvegarder les paramètres sous forme de document ***.rules*** dans les fichiers d'***iptables***.

Les commandes sont les suivantes :

iptables-save > /etc/iptables/iptables.rules

et

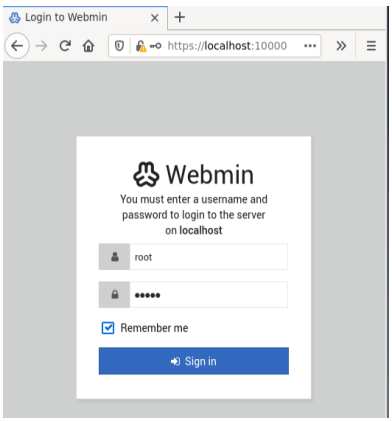
iptables-restore < /etc/iptables/iptables.rules

Vous voulez en savoir davantage et vous lancer ?

Je vous conseille ce site : ***https://inetdoc.net/guides/iptables-tutorial/***

3.3.2.4 C'est compliqué ? On peut utiliser ***webmin***, une belle interface graphique ☺

Tout est là pour l'installer : https://www.webmin.com/ qui écoute le port 10000



1. **Lancez *dpkg -i webmin_2.013_all.deb***
2. Si vous êtes en **sudo** et bien il faudra donner un MDP au compte root ☺

Remarque : ceci est un peu hors sujet au cours mais montre qu'une machine sous Linux peut faire beaucoup de choses...

Autre remarque : **Iptables** est remplacé à plus ou moins long terme par **Nftables**.

Un excellent TUTO pour ceux qui voudraient s'y lancer pour le Master e-secure

https://linuxembedded.fr/2022/06/introduction-a-nftables

4 Le port forwarding

Le ***port forwarding*** consiste à rediriger un port du routeur vers un port donné sur une machine locale ayant généralement une adresse IP privée.

Imaginons le scénario suivant :

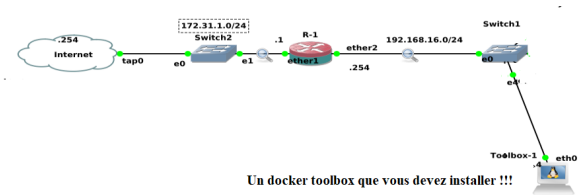
Un serveur avec 4 services,

1. **http** sur le port **80**,
2. **telnet** sur le port **23,**
3. **ssh** sur le port **22**
4. **ftp** sur le port **21**

est installé dans la **partie IP privée** d'une organisation.

Théoriquement, il est impossible d'y accéder de l'extérieur !

On va donc faire du **port forwarding** et au lieu de donner l'@IP Privée du serveur, on va donner l'IP Publique du routeur qui lui est visible sur Internet !



Pour installer cette machine qui est un Docker,

Allez dans le menu pour ajouter un composant comme si vous ajoutiez un routeur ou un VPPCS, en bas cliquez sur +New template, une nouvelle fenêtre apparait, dans les GUUEST, choisissez celuiqui se nomme toobox et cliquez sur install.

Il va s'installer automatiquement.

Quand vous allez la chercher pour la mettre dans le design, vous allez voir apparaittre en haut à Droite son "pulling", c'est à dire que l'on charge l'image du docker, comme on téléchage un fichier VDI, image de votre débian. Cela dure un cetain temps, le temps detélécharger une centaine de M0... oui ui une VM qui fonctionne avec un DD de 100Mo et qui fait tourner

****

J'ai volontairement fait un disable des 4 ports (21, 22, 23 et 80) des serveurs du routeur R1, car les routeurs Mikrotik écoutent ces 4 ports ! ! !

Comme le montre la figure suivante : Il est possible de faire un "***disable***" d'un port ! ! !

```
[admin@MikroTik] /ip service> print
Flags: X - disabled, I - invalid
 #   NAME       PORT ADDRESS
 0   telnet       23
 1   ftp          21
 2   www          80
 3   ssh          22
 4 XI www-ssl     443
 5   api        8728
 6   winbox     8291
 7   api-ssl    8729
[admin@MikroTik] /ip service> disable 0
[admin@MikroTik] /ip service> disable 1
[admin@MikroTik] /ip service> disable 2
[admin@MikroTik] /ip service> disable 3
[admin@MikroTik] /ip service> print
Flags: X - disabled, I - invalid
 #   NAME       PORT ADDRESS
 0 XI telnet      23
 1 XI ftp         21
 2 XI www         80
 3 XI ssh         22
 4 XI www-ssl     443
 5   api        8728
 6   winbox     8291
 7   api-ssl    8729
[admin@MikroTik] /ip service>
```

Configuration du routeur : Voici les règles de port ***forwarding***

```
[admin@R1] /ip firewall nat> print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=dstnat action=dst-nat to-addresses=192.168.16.4 to-ports=1664 protocol=tcp dst-address=172.31.1.1
      port=1664 log=no log-prefix=""
           80
 1    chain=srcnat action=masquerade out-interface=ether1

 2    chain=dstnat action=dst-nat to-addresses=192.168.16.4 to-ports=2222 protocol=tcp dst-address=172.31.1.1
      dst-port=2222 log=no log-prefix=""

 3    chain=dstnat action=dst-nat to-addresses=192.168.16.4 to-ports=2121 protocol=tcp dst-address=172.31.1.1
      dst-port=2121 log=no log-prefix=""
[admin@R1] /ip firewall nat>
```

Pour HTTP

Pour SSH

Pour FTP

Interprétons pour le http par exemple

Tout paquet IP qui arrive sur l'interface 172.31.1.1 avec le numéro de port 1664 du routeur est transféré à la machine ayant l'IP 192.168.16.4 avec le port 80.

Si on fait un "***disable***" du serveur Web du routeur, la commande est a modifier en changeant le second 1664 par 80.

Ce qui permet d'avoir un serveur Web avec une @IP privée...

Sur le **navigateur de la VM on saisit l'IP 172.31.1.1**, cette requête va sur le routeur qui va la transférer (***forwarding*** !) au serveur Web à l'adresse IP privée 192.168.16.4 sur le port 1664

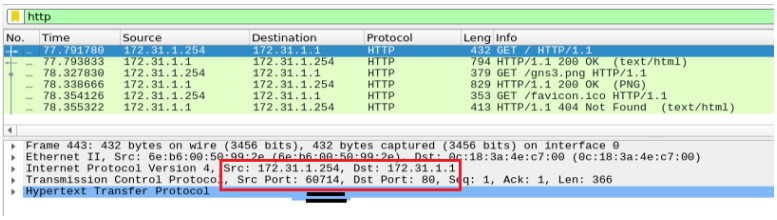
Comme on peut le voir sur les 2 captures wireshark suivantes

(ATTENTION, j'ai modifié le port d'écoute du serveur http nginx, ce n'est pas fait dans le TP ! ! !)

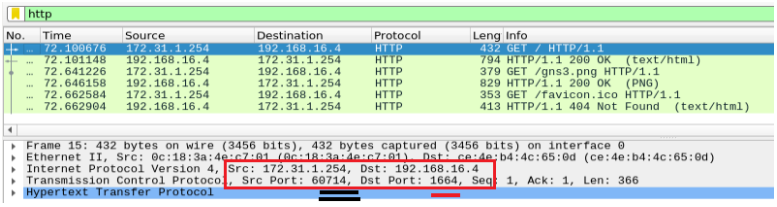
REMARQUEZ : le numéro de port **60714** qui est le même en **entrée** et en **sortie** du routeur ! ! !

Les IP et les Numéros de ports d'écoute change ! ! !

Avant le routeur



Après le routeur



5 Conclusion

— On peut avoir un serveur Web en IP privée comme tout autre serveur d'ailleurs !
— Quand vous êtes à l'intérieur vous utilisez les @IP privées et à l'extérieur les @IP publiques !
— Et le serveur physique lui n'a peut être voire sûrement aucune de ces 2 adresses ☺
— On peut avoir une adresse IP privée pour aller sur Internet.
— Vous devez retenir ce qu'est le **NAT** et le **Port Forwarding**.
— Nous verrons cela dans les 2 TP intitulé TP NAT-1 et TP NAT-2
— La partie IP Tables c'est de l'information pour ceux qui iraient en Master e-secure et/ou ceux qui voudraient s'y lancer