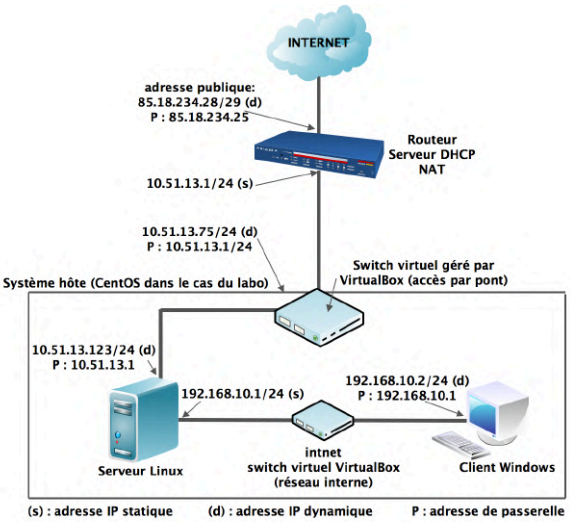
Partage de connexion Internet rédigé pour AlmaLinux 8.5

Hainaut Patrick 2022

But de cette présentation

- Maintenant que nous avons configuré le rôle serveur DHCP sur notre serveur Linux, donnons l'accès à Internet aux clients du réseau local
- Cela permettra de voir une première règle de firewall

Schéma de principe



Le serveur Linux est client DHCP sur l'interface réseau reliée au routeur (enp0s3 par exemple) et serveur DHCP sur l'autre (enp0s8 par exemple)

Le PC client reçoit ses paramètres IP du serveur Linux.

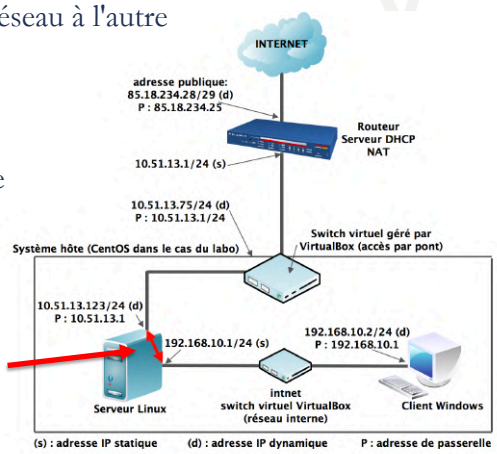
Partage de connexion Internet

- On voudrait pouvoir à partir du PC client, surfer sur le net …
- Pour cela, il va nous falloir configurer deux choses sur notre serveur Linux
  - Activer le passage des trames IP d'une interface réseau à l'autre
  - Faire du NAT (Network Address Translation)

Passage des trames IP d'une interface à l'autre

- Linux possède un interrupteur général autorisant ou non le passage d'une trame IP d'une carte réseau à l'autre (entre enp0s3 et enp0s8)

- Si l'interrupteur est à 'off ', les demandes en provenance du PC client, arrivant dans notre exemple sur enp0s8, ne seront jamais transmises à enp0s3, reliée au réseau extérieur



Passage des trames IP d'une interface à l'autre

- Cet interrupteur est contrôlé via la variable **ip_forward**

- **cat /proc/sys/net/ipv4/ip_forward** permet de voir l'état de l'interrupteur:
  0: pas de passage
  1: passage autorisé selon la configuration

- **echo 1 > /proc/sys/net/ipv4/ip_forward** permet d'activer cet interrupteur

- Au redémarrage du PC, ce réglage à disparu

Passage des trames IP d'une interface à l'autre

- Donc, pour changer la variable **ip_forward** durablement, vous pouvez ajouter la ligne suivante, à un endroit quelconque du fichier (ni tout au début, ni tout à la fin) : **net.ipv4.ip_forward = 1**

  dans le fichier **/usr/lib/sysctl.d/50-default.conf**

  Ce fichier sera lu par le fichier principal **/etc/sysctl.conf**

- Pour recharger la config.: **sysctl -p**

Firewall sous AlmaLinux

- Attention, sous AlmaLinux plusieurs firewall existent
  - firewalld
  - iptables
  - selinux

- C'est bien sur intéressant mais cela peut perturber nos futures manipulations et on préférera configurer son propre firewall (plus tard)

Firewall sous AlmaLinux

- Pour nos manipulations, on va utiliser iptables au lieu de firewalld
- Pour cela on désactive d'abord firewalld (si il est activé):

  systemctl stop firewalld
  systemctl disable firewalld (pour qu'il ne soit pas réactivé au reboot)

  si vous avez un message d'erreur qui vous indique que firewalld n'est pas actif, pas de soucis, c'est qu'il est désactivé par défaut

Firewall sous AlmaLinux

- Une autre protection est constituée par SELinux (Security Enhanced Linux)
- Pour les besoins de la manip, nous allons désactiver SELinux
- Pour cela, il faut modifier la ligne **SELINUX=enforcing** par **SELINUX=permissive** (restrictions logguées mais pas appliquées) ou **SELINUX=disabled** (rien n'est loggué ni appliqué) dans le fichier **/etc/selinux/config**
- Le nouveau réglage ne sera pris en compte qu'après reboot mais on peut taper la commande **/usr/sbin/setenforce 0** qui désactive temporairement SELinux

Firewall sous AlmaLinux

- On va installer iptables et iptables-services pour configurer le NAT

  dnf install iptables

- Il se peut que iptables soit déjà installé, dans ce cas, la commande se terminera sans installer de paquets

  dnf install iptables-services

- iptables-services va servir à sauvegarder les règles de firewall qu'on va taper, pour qu'elles soient encore actives après un reboot de l'OS

Firewall sous AlmaLinux

- On démarre iptables et on l'autorise à se réactiver après un reboot de l'OS

  systemctl start iptables
  systemctl enable iptables

- Un reboot de Linux s'impose, il suffit de taper la commande

  reboot

Firewall sous AlmaLinux

- Des règles de filtrage sont configurées par défaut dans iptables
- Pour les éliminer, il faut taper
  iptables --flush -t filter
  iptables --flush -t nat
- Remarques:
  - iptables comprend 3 tables: filter, mangle et nat
  - filter permet de décider des paquets qui passent à travers le firewall
  - mangle permet de transformer les paquets
  - nat permet de faire du nat
  - -t filter est ici optionnel car la table filter est la table par défaut

Principe du NAT dynamique

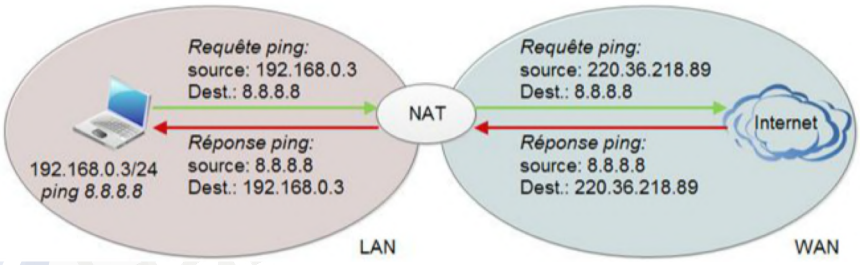
- Pour surfer sur Internet, il faut une adresse IP publique
- Dans un réseau local, en général, une seule machine est reliée à Internet, les autres PC passant par elle (on l'appelle d'ailleurs 'passerelle' ou 'gateway' en anglais) pour sortir
- Cette passerelle sera généralement constituée par un routeur ou un serveur assumant la fonction de routage
- Un PC du réseau local passant par cette passerelle ne pourra pas accéder directement à Internet car son adresse IP privée ne lui permet pas

Principe du NAT dynamique

- Il faut donc un mécanisme qui va 'échanger' l'adresse IP privée du PC client par une adresse IP publique (celle de la passerelle)
- C'est le mécanisme de translation d'adresse (NAT en anglais)
- Quand la passerelle revient avec les données d'Internet, elle échange de nouveau les adresses pour transmettre ces données au PC client
- La passerelle contient une table dynamique qui lui permet de savoir qui est à l'origine de quelle requête et donc remettre la réponse au bon destinataire sur le réseau local

Principe du NAT dynamique

- Tout le réseau local peut donc "surfer" sur Internet avec une seule machine effectivement connectée à Internet (la passerelle)

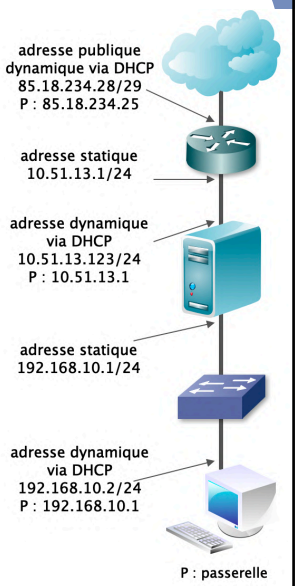


Principe du NAT dynamique

- En plus de faire de la translation d'adresses, le NAT dynamique utilise le mécanisme de translation de port (**PAT** - *Port Address Translation*)
- Cela affecte un port source différent à chaque requête de telle manière à pouvoir maintenir une correspondance entre les requêtes provenant du réseau interne et les réponses des machines sur Internet, toutes adressées à l'adresse IP du routeur.

Principe du NAT dynamique

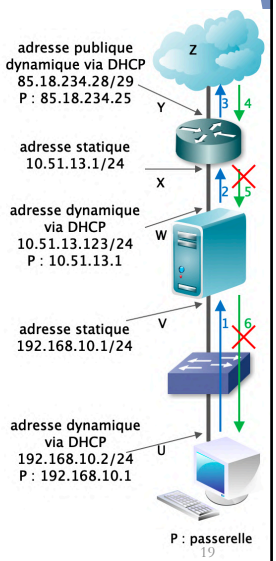
- Dans notre cas, l'adresse IP du serveur Linux du coté WAN est une adresse privée (dans notre exemple: 10.51.13.123)
- Théoriquement, on ne doit donc pas mettre en œuvre le NAT sur ce serveur
- Le NAT sera mis en œuvre sur le routeur
- Pourtant, si vous faites un ping d'une adresse publique (8.8.8.8), ça ne fonctionne pas …



Principe du NAT dynamique

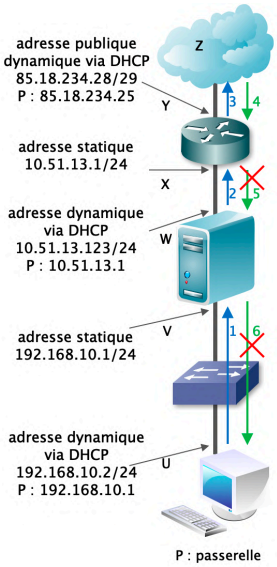
- Examinons les trames Ethernet échangées …

	Trame Ethernet		Paquet IP		
1	MAC des: V	MAC src: U	IP des:8.8.8.8	IP src:192.168.10.2	Datas
2	MAC des: X	MAC src: W	IP des:8.8.8.8	IP src:192.168.10.2	Datas
3	MAC des: Z	MAC src: Y	IP des:8.8.8.8	IP src:85.18.234.28	Datas
4	MAC des: Y	MAC src: Z	IP des:85.18.234.28	IP src:8.8.8.8	Datas
5	MAC des: ?	MAC src: X	IP des:192.168.10.2	IP src:8.8.8.8	Datas
6	MAC des:	MAC src:	IP des:	IP src:	Datas



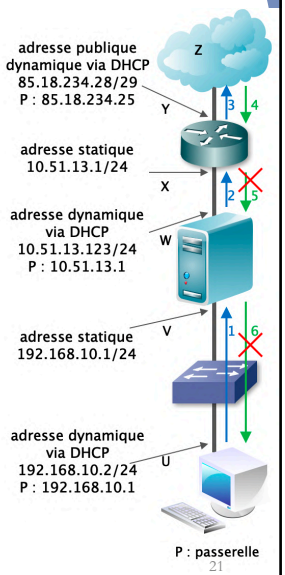
Principe du NAT dynamique

- Pour l'aller, pas de problème
- Mais pour le retour, le routeur qui applique le NAT dans l'autre sens, et qui construit donc une trame Ethernet avec 192.168.10.2 comme IP de destination, n'a pas de route dans sa table de routage pour le réseau 192.168.10.0/24 …
- Il a une route par défaut vers Internet et une autre vers le réseau 10.51.13.0/24
- Il laisse donc tomber le paquet IP



Principe du NAT dynamique

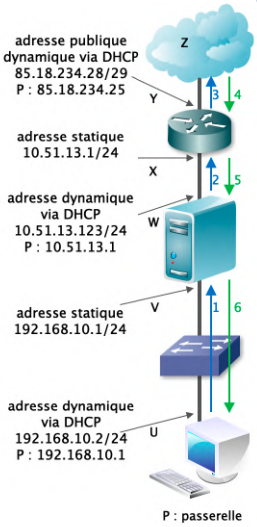
- Pour que ça puisse fonctionner, deux méthodes:
  - Soit on rajoute une route dans le routeur vers le réseau 192.168.10.0/24
  - Soit on rajoute une règle de NAT sur le serveur Linux
  - La première solution est plus professionnelle (car le NAT est prévu pour échanger une IP privée par une IP publique, ce qui ne sera pas le cas ici) mais on doit avoir accès au routeur …



Principe du NAT dynamique

- Si on active le NAT sur le serveur Linux …

	Trame Ethernet		Paquet IP		
1	MAC des: V	MAC src: U	IP des:8.8.8.8	IP src:192.168.10.2	Datas
2	MAC des: X	MAC src: W	IP des:8.8.8.8	IP src:10.51.13.123	Datas
3	MAC des: Z	MAC src: Y	IP des:8.8.8.8	IP src:85.18.234.28	Datas
4	MAC des: Y	MAC src: Z	IP des:85.18.234.28	IP src:8.8.8.8	Datas
5	MAC des: W	MAC src: X	IP des:10.51.13.123	IP src:8.8.8.8	Datas
6	MAC des: U	MAC src: V	IP des:192.168.10.2	IP src:8.8.8.8	Datas



Mise en œuvre du Nat

- C'est notre serveur Linux qui jouera le rôle de passerelle
- On active le NAT (ou masquerading) par une règle de firewall:
  iptables –t nat –A POSTROUTING –o enp0s3 –j MASQUERADE
- enp0s3 est l'interface reliée à Internet (adaptez suivant votre configuration)
- Pour que le réglage perdure après redémarrage, il faut sauver les règles tapées par la commande
  service iptables save

Mise en œuvre du Nat

- Faites un reboot de votre Linux et testé si les règles de firewall écrites sont toujours actives
- **iptables –t nat –L** affiche les règles pour la table NAT; vous devriez avoir la mention MASQUERADE au niveau de la chaine Postrouting
- **iptables –t filter –L (**ou **iptables – L)** affiche les règles pour la table filter; vous devriez avoir les 3 chaines en ACCEPT

```
[root@localhost ~]# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@localhost ~]# _
```

```
[root@localhost ~]# iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@localhost ~]#
```

Test sur le PC client

- Dans une invite de commande, un **ping** vers une adresse externe comme 8.8.8.8 (DNS de Google) devrait passer
- Un **ping** vers www.google.be devrait fonctionner aussi pour peu qu'un serveur DNS correct soit spécifié dans la configuration de notre serveur DHCP
- Si ce n'est pas le cas, cherchez du coté des firewalls (serveur et client)
  - Vérifiez que firewalld et selinux sont bien désactivés
  - Vérifiez que votre règle iptable de NAT soit correcte et bien sauvegardée, au besoin faîtes un flush et recommencez
  - Vérifiez que la variable ip_forward soit bien à 1

Serveur DNS

- C'est la directive
  option domain-name-servers <IP du serveur DNS>;

  du fichier **/etc/dhcp/dhcpd.conf** qui permet de spécifier l'adresse du serveur DNS

  Ex.: **option domain-name-servers 8.8.8.8;**

  Cela oblige à connaître l'IP d'un serveur DNS externe … et si on héberge des sites internes, ceux-ci ne seront pas pris en compte

Installer son propre serveur DNS

- Si vous voulez que les requêtes DNS du client passe par votre serveur Linux, vous pouvez installer les paquets **bind** (le serveur proprement dit) et **bind-utils** (des utilitaires comme dig)
- L'adresse du serveur DNS à spécifier dans la config du serveur DHCP (/etc/dhcp/dhcpd.conf) est maintenant celle du serveur Linux
- Dans notre exemple, cela donne:
  option domain-name-servers 192.168.10.1;

Configuration de Bind

- La configuration de Bind se fait au travers de plusieurs fichiers:
  - /etc/named.conf pour la configuration globale
  - /etc/named.rfc1912.zones pour déclarer les zones du domaine
- Ici, on veut pour l'instant que notre serveur DNS serve de relais DNS vers un serveur DNS externe comme celui de Google (8.8.8.8)

Configuration de Bind

- On va modifier le fichier /etc/named.conf

  Les lignes à modifier sont:

  listen-on port 53 { 127.0.0.1; *192.168.10.1;* };
  avec 192.168.10.1 l'adresse de mon serveur Linux, aussi serveur DNS

  allow-query { localhost; *192.168.10.0/24;* };
  avec 192.16810.0 le "range" IP du réseau local

  mais aussi:

  dnssec-enable *no*; et dnssec-validation *no*;
  (qui étaient à *yes* par défaut)

Configuration de Bind

- La deuxième ligne autorise le réseau local à interroger le serveur DNS (et seulement lui)

- Les deux dernières lignes concernent la sécurité du serveur DNS et on verra comment réactiver ces lignes lorsqu'on abordera les questions de sécurité …

Configuration de Bind

- Pour vérifier qu'il n'y a pas d'erreurs de syntaxe dans un des fichiers de configurations, vous pouvez taper la commande:

  named-checkconf –z

- Après les modifications, il faut redémarrer le serveur DNS, ce qui ce fait en tapant la commande:

  systemctl start named

- Pour qu'il soit activé au démarrage, il faut taper la commande:

  systemctl enable named

Test sur le serveur

- Pour vérifiez que votre serveur DNS fonctionne, vous pouvez utiliser la commande **dig**:

  dig *@adresse_du_serveur_DNS cible*

  Exemple: **dig** @192.168.10.1 www.google.be

- Vous devriez avoir une réponse à votre requête vous donnant l'adresse IP correspondant à www.google.be dans cet exemple

- Si ça ne fonctionne pas, vérifiez la syntaxe de vos fichiers de configuration DNS et que votre serveur à toujours accès à Internet

Test sur le PC client

- Après avoir renouvelé les paramètres IP (**ipconfig /release** suivi de **ipconfig /renew**), testez l'accès au réseau extérieur (**ping 8.8.8.8**) et la résolution de noms (**ping www.google.be**)
- Ouvrez votre navigateur Internet favori et surfez … (n'oubliez pas de renseigner le proxy éventuel …)

Test sur le PC client

- A noter que ces manipulations peuvent fonctionner aussi avec un PC physique ou un réseau physique relié à une carte réseau réelle placée dans le PC (enp0s8 dans notre exemple)
- Vérifiez quand même qu'il n'y ait qu'une seule passerelle de sortie …

Sécurisation du serveur DNS

- En prenant quelques mesures basiques, nous allons augmenter la protection de notre serveur DNS contre deux attaques courantes:
  - Le DNS Spoofing qui consiste à se faire passer pour le serveur DNS et envoyer des réponses erronées de manière à rediriger l'internaute vers un serveur pirate pour par exemple enregistrer des données sensibles
  - Le DNS cache poisoning qui consiste à remplir le cache du serveur DNS par des données erronées de manière à rediriger l'internaute vers un serveur pirate ou vers un site web falsifié (fishing)

Restrictions sur le transfert de zone

- On va interdire le transfert de zone afin qu'un serveur DNS inconnu ne puisse pas venir lire les informations sur les zones DNS de notre serveur
- Pour cela, il faut éditer le fichier **named.conf** et ajouter, dans la section options, la ligne **allow-transfer {none;};**

Masquer la version du serveur DNS

- On veillera à maintenir la version de son serveur DNS à jour pour se prémunir contre les failles de sécurité découvertes
- Et pour éviter de faciliter la tâche des pirates, on va éviter de diffuser la version de son serveur DNS
- Si vous tapez la commande:
  dig @adresse_serveur_DNS version.bind txt chaos sous linux, le système vous renvoie la version du serveur DNS
  Exemple: **dig @192.168.10.1 version.bind txt chaos**
- Pour éviter cela, il faut éditer le fichier **named.conf** et ajouter, dans la section options, la ligne **version "DNS";**

Fichier named.conf.options résultant

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {8.8.8.8;};

        allow-query {192.168.10.0/24;};

        allow-transfer {none;};

        version "test";

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { none; };
        listen-on {127.0.0.1; 192.168.10.1;};
};
```

Fichier named.conf.options résultant

- N'oubliez pas de redémarrer le serveur DNS après une modification d'un des fichiers de configuration (**service named restart**)
- Si vous avez des soucis, vérifiez la syntaxe (**named-checkconf –z**)

Conclusion

- Notre serveur Linux sert donc maintenant de passerelle vers Internet, tout en étant serveur DHCP et DNS
- Ces rôles auraient pu être tenus par un routeur multifonction, mais notre serveur pourra encore tenir d'autres rôles, comme celui de serveur de fichiers, ou de contrôleur de domaine …
- Merci de votre attention