Leçon 6 : Configuration du routage

6.1 Introduction

Dans cette leçon, nous allons découvrir comment il est possible de **transformer sa machine Linux en routeur**. Cette caractéristique est très intéressante et souvent utilisée. Un système Linux est capable de se substituer à n'importe quel routeur : muni d'interface réseau rapide, il peut serveur de véritable firewall pour une entreprise, de passerelle intelligent entre plusieurs réseaux ou encore de véritable routeur implémentant des algorithmes nombreux comme OSPF ou BGP.

Loin de toute cette complexité, nous allons découvrir dans cette leçon comment un serveur Linux peut servir de *passerelle* afin de sortir du réseau.

6.2 Rappel : un routeur

Un routeur finalement, **c'est quoi ?** Nous pourrions répondre de manière très complexe à cette question, disons simplement que, dans le cadre de ce cours, un routeur est une machine :

1. Disposant de plusieurs interfaces réseaux interconnectées à des sous-réseaux distincts
2. Configurée pour que le trafic réseau puisse passer d'une interface à l'autre

Ainsi, à la différence d'une machine « normale » (ne jouant pas le rôle de routeur), elle reçoit le trafic lorsqu'elle est destinataire de celui-ci mais jamais elle ne propage des paquets d'une interface réseau vers une autre.

Avant de continuer, il est nécessaire de rappeler l'importance de **la table de routage** présente sur chaque machine. En IPv4, cette table est obtenue comme suit :

```
$ ip route show
default via 192.168.190.2 dev eno16777736  proto static  metric 100
10.0.1.0/24 dev eno16777736  proto kernel  scope link  src 10.0.1.2  metric 100
192.168.190.0/24 dev eno16777736  proto kernel  scope link  src 192.168.190.50
metric 100
```

En lisant ces informations, nous apprenons que :

- La route par défaut, nommée `default` (ou parfois `0.0.0.0/0`), utilise la passerelle 192.168.190.2 (`via 192.168.190.2`) sur l'interface réseau eno16777736 (`dev eno16777736`). Donc la route par défaut utilise la passerelle pfSense pour sortir du réseau.
- Le réseau `10.0.1.0/24` est connecté directement (`scope link` – absence d'option `via`) à l'interface réseau eno16777736 (`dev eno16777736`)
    - Donc un `ping` vers `10.0.1.5` sera transmis directement sur le réseau sans être adressé au routeur de sortie pfSense
- Le réseau `192.168.190.0/24` est connecté directement (`scope link` – absence d'option `via`) à l'interface réseau eno16777736 (`dev eno16777736`)
    - Donc un `ping` vers `192.168.190.1` sera transmis également directement sur le réseau sans être adressé au routeur de sortie pfSense

A l'inverse, un `ping` vers `8.8.8.8` passera par la route par défaut configurée[21] et l'information sera alors adressée à pfSense pour sortir du réseau virtuel.

6.3 Configuration du routeur Linux

Supposons que nous désirons configurer le réseau comme suit :

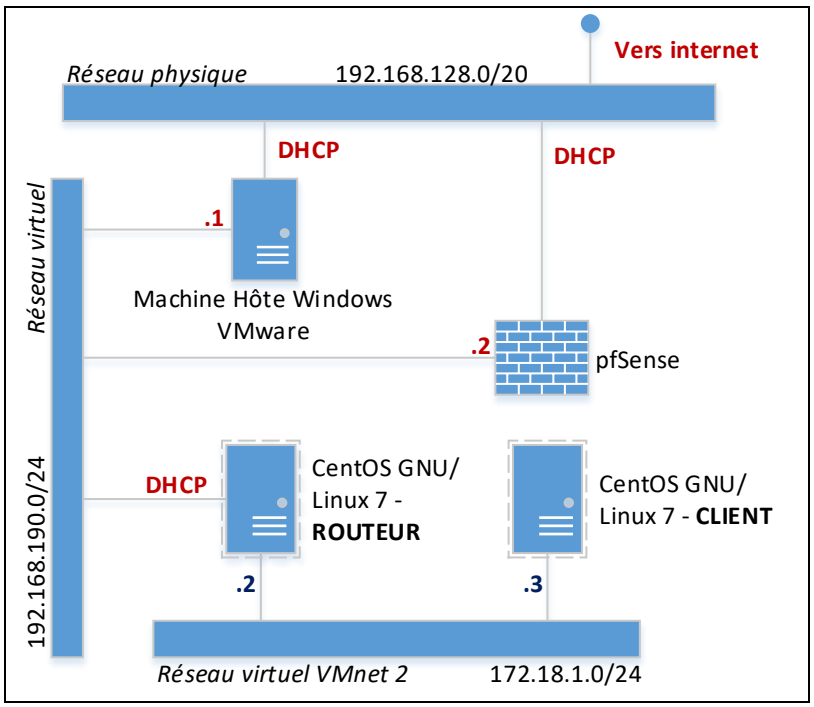

Figure 6.1 : Réseau virtuel souhaité

La *figure 6.1* montre le réseau virtuel que nous souhaitons configurer. Ainsi, nous voyons que la machine *CentOS7 Routeur* est connectée à 2 réseaux : l'actuel réseau virtuel et un second (nommé VMnet 2). Nous avons également ajouté une seconde machine CentOS 7, nommée *CentOS 7 Client* qui est uniquement connectée au nouveau réseau virtuel. Pour atteindre internet, cette machine doit donc passer par la machine *CentOS7 Routeur*.

Nous supposerons donc dans la suite de cette leçon[22] que :

1. La machine *Routeur* dispose de 2 interfaces réseaux : `eno16777736` connectée au réseau virtuel `192.168.190.0/24` et `eno33554976` connectée au réseau virtuel `172.18.1.0/24`.
2. La machine *Client* dispose d'1 interface réseau : `eno16777736` connectée au réseau virtuel `172.18.1.0/24`.
3. Les adresses IP des machines *Routeur* et *Client* sont bien configurées (il faut se référer à la leçon sur la configuration du réseau pour y arriver – le routeur utilise l'adresse `172.18.1.2` et le *client* utilise l'adresse `172.18.1.3`)
4. La machine *client* utilise la machine *routeur* comme route par défaut.

[21] Puisque cela ne correspond ni au réseau `192.168.190.0/24` ou `10.0.1.0/24`.
[22] Nous reviendrons sur les modifications à apporter dans VMware pour atteindre cette connexion réseau

6.3.1 Configurer la machine en mode routeur

Pour configurer la machine *CentOS 7 Routeur* en mode « routeur », il faut ajouter un fichier dans le dossier `/etc/sysctl.d` :

```
$ vim /etc/sysctl.d/10-ipforward.conf
# Enabling IP Forwarding
net.ipv4.ip_forward=1
```

Cette configuration s'assure, **qu'au démarrage du système Linux**, celui-ci autorise les paquets à passer d'une interface à l'autre (`eno16777736` ⇔ `eno33554976`).

Il est possible d'activer ce mode directement en entrant :

```
$ echo 1 > /proc/sys/net/ipv4/ip_forward
```

En utilisant cette commande, le mode « routeur » est activé temporairement jusqu'au redémarrage de la machine, à moins que la configuration précédente (ajout du fichier `10-ipforward.conf`) ait été faite également.

Problème

Nous remarquons qu'après l'activation du mode « routeur », la machine *Client* ne peut toujours pas atteindre Internet. Cependant, si nous analysons le trafic reçu par la machine *Serveur* nous voyons qu'il est bien reçu et propagé sur l'autre interface. La machine semble donc bien jouer son rôle de routeur.

Où se situe le problème alors ?

Le problème vient simplement du fait que le réseau `172.18.1.0/24` n'est pas connu plus loin. Puisque **les paquets sont propagés sans modification**, ils atteignent le réseau de l'école mais **comment la réponse peut-elle parvenir ?**

En effet, ni le réseau de l'école, ni pfSense ne connait `172.18.1.0/24`. Dès lors, ils sont incapables de transmettre la réponse à notre machine *Client*.

Il y a 2 moyens de solutionner le problème :

- Configurer pfSense pour qu'il connaisse le réseau `172.18.1.0/24` en ajoutant *une route statique* pointant vers la machine *CentOS Routeur*. Ensuite, il faudrait autoriser ces paquets à aller sur internet dans la configuration du firewall.
- Configurer la machine *CentOS 7 Routeur* pour qu'elle translate les adresses (activation du NAT). Si nous adoptons cette solution, la machine *CentOS 7 Routeur* remplacera l'adresse IP source `172.18.1.2` par son adresse IP `192.168.190.x`, rendant ainsi invisible le réseau virtuel 2. C'est cette solution que nous allons configurer.

6.3.2 Activation du NAT sur la machine Routeur

Comme indiqué dans le paragraphe précédent, activer le NAT sur la machine *Routeur* est une solution au problème soulevé. Pour ce faire, il faut introduire une toute petite règle dans le firewall *iptables* de la machine *Routeur*.

Nous aurons, lors d'une prochaine leçon, l'opportunité d'étudier en profondeur le fonctionnement d'*iptables*. Nous allons ici simplement discuter de la règle à configurer :

```
$ iptables -t nat -A POSTROUTING -s 172.18.1.0/24 -j MASQUERADE
```

Cette commande ajoute dans le système NAT une règle indiquant que le trafic dont l'IP source est comprise dans le sous-réseau `172.18.1.0/24` (`-s 172.18.1.0/24`) doit être translaté (`-j MASQUERADE`).

Une fois la règle introduite, la machine *Client* devrait pouvoir accéder à Internet. En cas de problème il convient de vérifier la configuration réseau, conformément aux directives de la leçon précédente.

Afin de rendre cette configuration permanente, il faut sauvegarder la règle de firewall de sorte que celle-ci soit automatiquement ajoutée lors du lancement de la machine *Routeur*. Pour ce faire, il faut simplement entrer :

```
$ service iptables save
```

Cette commande sauvegarde la configuration actuelle du firewall iptables de sorte que celle-ci soit rechargée automatiquement lors du démarrage de la machine.

6.4 Quelques outils réseaux

Nous allons décrire ici quelques outils réseaux intéressants. **Parmi ceux-ci, il y a *nmap*, un outil qui permet notamment de *scanner* les ports ouverts et *hping*, un outil qui permet de construire des paquets IP. Utiliser de tels outils sur des réseaux étrangers PEUT ÊTRE CONSIDERÉ COMME UN ACTE HOSTILE. Il faut toujours restreindre l'utilisation de ces outils sur des réseaux que vous gérez et configurez.**

6.4.1 wireshark

Le programme *wireshark* permet de capturer et analyser le trafic réseau. Il est particulièrement utile pour trouver les problèmes qui surviennent. Dans notre configuration réseau actuelle, lancer *wireshark* sur la machine *CentOS 7 Routeur* permet de capturer tout le trafic venant de la machine *CentOS 7 Client* de manière discrète.

Afin de voir uniquement le trafic qui vous intéresse, vous pouvez définir des filtres. Reportez-vous à la documentation pour la construction de ceux-ci.

Pour le démarrer :

```
$ wireshark
```

6.4.2 NMap

Le programme *nmap* est une boite à outils réseaux. Il peut être utilisé pour identifier le système distant, lister les ports ouverts, … **Comme annoncé en introduction, cet outil doit être réservé à des réseaux que vous gérez**.

Quelques commandes intéressantes :

```
$ nmap -A -T4 127.0.0.1
```

Cette commande lance l'analyse en mode agressif (-A et -T4). Le résultat donne la liste des ports ouverts sur la machine visée (ici 127.0.0.1), une identification du système d'exploitation et de la version. L'option -A est à déconseiller si l'on veut rester discret. L'option -T détermine le comportement (T0 – Paranoid, T1 – Sneaky, T2 – polite, T3 – normal, T4 – aggressive, T5 – Insane).

```
$ nmap 192.168.190.2
```

Celle commande analyse les ports ouverts (du coté LAN) du firewall pfSense.

```
$ nmap -sP 172.18.1.0/24
```

Cette commande liste toutes les adresses IP actives en utilisant des requêtes ICMP. Attention, beaucoup de firewall bloquent ces requêtes.

```
$ nmap -PS80 172.18.1.0/24
```

Cette commande liste toutes les adresses IP actives en utilisant des demandes de connexion sur le port 80. Nous obtiendrons ainsi la liste des machines exécutant un serveur web.

Il existe également une version graphique de *nmap* :

```
$ nmapfe
```

6.4.3 Telnet

Telnet est un outil simple : il permet d'ouvrir une connexion TCP sur n'importe quel port et puis interagir avec le serveur facilement. Par exemple :

```
$ telnet mail.helmo.be 110
+OK Dovecot ready.
```

Cette commande permet d'ouvrir une connexion TCP vers le serveur POP3 (port 110) de HELMo. Une fois l'invite affiché (+OK Dovecot ready), il est possible d'introduire des commandes POP3 pour obtenir des réponses (`USER/PASS/LIST/RETR/DELE/QUIT`).

Cette méthode permet souvent de connaître le type de serveur installé (ici Dovecot) et parfois la version déployée. Ces informations sont nécessaires pour déterminer si des vulnérabilités sont présentes dans les logiciels installés.

```
$ telnet localhost 22
SSH-2.0-OpenSSH_6.6.1
```

Ici nous remarquons que le service SSH utilise la version OpenSSH 6.6.1. On peut faire de même avec le service FTP :

```
$ telnet ftp.belnet.be 21
220 ProFTPD 1.3.4a Server (Belnet FTP Server) [193.190.67.98]
```

Il faut donc être prudent lorsqu'on configure un service réseau : il faut s'assurer qu'il ne diffuse pas d'information d'identification le concernant.

6.4.4 HPing

HPing est une autre boîte à outils réseau. Grâce à hping, il est possible de construire des paquets TCP de toute pièce.

hping permet notamment de lancer *une commande ping* en utilisant TCP/IP plutôt que ICMP (qui est souvent bloqué par les firewalls).

```
$ hping -S 192.168.190.2 -p 80
HPING 192.168.190.2 (eno16777736 192.168.190.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.190.2 ttl=64 DF id=31862 sport=80 flags=SA seq=0 win=65228
rtt=0.4 ms
```

Cette commande envoie des paquets TCP sur le port 80 (-p 80). Si la réponse est flags=SA, c'est que le port en question est ouvert. Dans le cas contraire, la réponse flags=RA est rapportée.

Il est également possible d'utiliser *hping* pour *scanner les ports ouverts* d'une machine (-p ++80).

Il est même possible d'utiliser *hping* pour échanger des fichiers entre deux machines de manière très discrète, par exemple en utilisant des paquets ICMP :

Emetteur	Destinataire
`$ hping 172.18.1.2 --icmp -d 100 --sign monfichier --file /etc/passwd`	`$ hping --listen 172.18.1.2 -I eno33554976 --sign monfichier --icmp`

Il est également possible d'effectuer un *traceroute* en utilisant *hping*. L'intérêt étant de pouvoir utiliser **n'importe quel paquet TCP** (au lieu des traditionnels paquets UDP ou ICMP).

Par exemple, avec la commande :

```
$ hping -z -t 1 -S www.swila.be -p 80
```

Cette commande permet de lancer des connexions sur le port TCP 80 avec des TTLs croissants en commençant à 1 (`-t 1`) vers la destination www.swila.be (option `-S`). Chaque fois que l'utilisateur appuie sur `CTRL+Z`, hping incrémente le TTL (option `-z`) et tente de résoudre l'adresse IP obtenue.

L'outil *hping* a encore bien d'autres possibilités comme lancer des attaques DoS vers un serveur (via l'option `--flood`). Je vous renvoie aux pages de manuels pour plus de détails.

6.5 Exercices

On vous demande de :

1. Eteindre votre machine *Linux CentOS 7*.
    - Modifier ses paramètres (*Edit virtual machine settings*) pour lui ajouter une nouvelle interface réseau (onglet *hardware*, Cliquer sur *Add* puis *Network Adapter*)
    - Choisir *Network Connection Custom => VMnet2* puis *Finish*
    - Onglet *Options* > Changer le *virtual machine name* (à droite) en *CentOS7-Routeur*
    - Démarrer la machine modifiée. La nouvelle interface réseau devrait pouvoir être configurée.

2. Ajouter une nouvelle machine CentOS 7 :
    - Décompresser le fichier contenant la machine CentOS **dans un nouveau dossier** (attention de ne pas écraser votre machine actuelle)
    - Ajouter cette machine dans VMware
    - Modifier ses paramètres (Edit virtual machine settings)
        i. Pour Network Adapter, choisir Network Connection *Custom > VMnet2*
        ii. Onglet Options > Changer le *Virtual machine name* en *CentOS7-Client*
    - Démarrer la nouvelle machine virtuelle – répondez *I copied it* à la question posée par VMware.

3. Configurer votre machine *CentOS 7 – Routeur* en mode « routeur ». L'adresse IP à configurer sur la 2ème interface réseau est `192.168.131.2`

4. Configurer votre machine *CentOS 7 – Client.* L'adresse IP à configurer est `192.168.131.15`. Cette machine doit utiliser la machine *routeur* comme passerelle. N'oubliez pas de mentionner un serveur DNS.

   Essayez de surfer à partir de la seconde machine. Utilisez *wireshark* pour capturer le trafic qui transite par la machine *routeur*.

5. A l'aide de *nmap,* lister tous les ports ouverts sur la machine *CentOS – Routeur* et *CentOS – Client*.

6. A l'aide de *hping,* simulez un *traceroute* sur le port 80 vers `www.yahoo.fr`

7. A l'aide de *nmap*, Vérifiez les ports ouverts sur la machine de votre voisin

8. Utiliser *hping* pour échanger un fichier texte entre vos 2 machines.