



# **SERVICES RESEAUX**

Armel KEUPONDJO

# PLAN

- GESTION DES HOTES
- LES SERVICES
- LES ATTAQUES
- LA SECURITE DES SERVICES

# GESTION DES HOTES

## ❑ Configuration d'une station

Un ordinateur communique avec les autres ordinateurs par des interfaces réseaux. En général, on a une interface réseau pour chaque carte réseaux (pour les ordinateurs qui ont plusieurs cartes réseaux). Une carte réseau (ethernet ou wifi) possède une adresse MAC, qui identifie la carte réseau sur le réseau. Pour faire fonctionner une carte réseau, il faut configurer l'interface qui li correspond.

# GESTION DES HOTES

## ❑ Configuration d'une station

La principale commande permettant de configurer le réseau est la commande **ifconfig**. Comme son nom l'indique (« InterFace CONFiguration »), elle permet de configurer les interfaces réseau de la machine. Il faut savoir qu'il existe plusieurs types d'interfaces réseau. Les plus courants sont les trois types d'interfaces suivants :

- ✓ l'interface loopback, qui représente le réseau virtuel de la machine, et qui permet aux applications réseau d'une même machine de communiquer entre elles même si l'on ne dispose pas de carte réseau ;
- ✓ les interfaces des cartes réseau (que ce soient des cartes Ethernet, TokenRing ou autres) ;
- ✓ les interfaces ppp, plip ou slip, qui sont des interfaces permettant d'utiliser les connexions sérieelles, parallèles ou téléphoniques comme des réseaux.

# GESTION DES HOTES

## ❑ Configuration d'une station

La configuration d'une interface comprend l'initialisation des pilotes nécessaires à son fonctionnement et l'affectation d'une adresse IP à cette interface. La syntaxe générale que vous devrez utiliser est la suivante :

*ifconfig interface adresse netmask masque up*

Où **interface** est le nom de l'interface réseau que vous voulez configurer, **adresse** est l'adresse IP que cette interface gèrera, et **masque** est le masque de sous-réseau que vous utilisez. Les interfaces que vous aurez à configurer seront certainement des interfaces Ethernet, auquel cas vous devrez utiliser les noms **eth0**, **eth1**, etc. dans la commande **ifconfig**. Si vous désirez configurer l'interface **loopback**, vous devrez utiliser le nom d'interface **lo**.

# GESTION DES HOTES

## ❑ Configuration d'une station

Le paramètre **up** donné à **ifconfig** lui indique que l'interface doit être activée. Cela signifie que dès que la commande **ifconfig** s'achèvera, votre interface réseau sera active et fonctionnelle. Bien entendu, il existe le paramètre inverse : **down**. Ce paramètre s'utilise tout simplement dans la commande **ifconfig** avec la syntaxe suivante :

***ifconfig interface down***

où **interface** est toujours le nom de l'interface.

Un exemple de configuration très classique est le suivant :

***ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up***

# GESTION DES HOTES

## ❑ Configuration d'une station

Le noyau utilisera par défaut le nombre **255** pour les adresses de broadcast dans les composantes de l'adresse IP qui ne fait pas partie de l'adresse de sous-réseau. Si vous désirez utiliser une autre adresse (en général, l'alternative est de prendre l'adresse du sous-réseau), vous devrez utiliser l'option broadcast adresse dans la commande **ifconfig**. Cependant, le comportement par défaut convient à la plupart des réseaux. La commande de configuration donnée en exemple ci-dessus sera alors :

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.0 up
```

# GESTION DES HOTES

## ❑ Configuration d'une station

Enfin, il est possible d'affecter plusieurs adresses IP à certaines interfaces réseau. C'est en particulier le cas pour toutes les interfaces réseau classiques, mais bien entendu cela n'est pas réalisable avec les interfaces de type point à point comme les interfaces des connexions ppp. Lorsqu'une interface dispose de plusieurs adresses, la première est considérée comme **l'adresse principale de l'interface**, et les suivantes comme **des alias**. Ces alias utilisent comme nom le nom de l'interface réseau principale et le numéro de l'alias, séparés par deux points (caractère ':'). Par exemple, si l'interface **eth0** dispose d'un alias, celui-ci sera nommé **eth0:0**. Ainsi, pour fixer l'adresse d'un alias d'une interface réseau, on utilisera la syntaxe suivante :

***ifconfig interface:numéro add adresse netmask masque***



# GESTION DES HOTES

## ❑ Configuration d'une station

Lorsque l'on n'a pas d'interface graphique ou lorsque l'on a désactivé Network-Manager on peut configurer le réseau en CLI directement dans des fichiers.

Depuis Ubuntu **17.10** et donc pour la dernière LTS en date **Ubuntu 18.04** la commande **ifconfig** n'est plus présente par défaut, le fichier **/etc/network/interfaces** n'est plus utilisé.

**Netplan** est un nouvel outil de configuration réseau qui utilise des fichiers de description YAML. Il vient avec **systemd / networkd**.

Tout est centralisé dans **/etc/netplan**

# GESTION DES HOTES

## ❑ Configuration d'une station

network:

ethernets:

eth0:

addresses: [192.168.1.200/24]

gateway4: 192.168.1.1

nameservers:

addresses: [8.8.8.8, 8.8.4.4]

dhcp4: false

dhcp6: false

version: 2

# GESTION DES HOTES

## ❑ Configuration d'une station

### ➤ Appliquer des changements

*netplan apply*

### ➤ Relancer le service

*systemctl restart systemd-networkd*

# GESTION DES HOTES

## ❑ Définition des règles de routage

La deuxième étape dans la configuration du réseau est la définition **des règles de routage**. Il est possible de définir plusieurs règles de routage actives simultanément. L'ensemble de ces règles constitue ce qu'on appelle la **table de routage**. La règle utilisée est sélectionnée par le noyau en fonction de l'adresse destination du paquet à router. Chaque règle indique donc un critère de sélection sur les adresses, et l'interface vers laquelle doivent être transférés les paquets dont l'adresse destination vérifie cette règle.

La commande utilisée pour définir une route est, chose surprenante, la commande système route. Sa syntaxe est donnée ci-dessous :

***route opération [-net | -host] adresse netmask masque interface***

# GESTION DES HOTES

## ❑ Définition des règles de routage

*route opération [-net | -host] adresse netmask masque interface*

où **opération** est l'opération à effectuer sur la table de routage. L'opération la plus courante est simplement l'ajout d'une règle de routage, auquel cas **add** doit être utilisé. L'option suivante permet d'indiquer si le critère de sélection des paquets se fait sur l'adresse du réseau destination ou plus restrictivement sur l'adresse de la machine destination. En général, il est courant d'utiliser la sélection de toutes les adresses d'un même réseau et de les router vers une même interface. Dans tous les cas, *adresse* est l'adresse IP de la destination, que celle-ci soit un réseau ou une machine. Si la destination est un réseau, il faut indiquer le *masque de sous-réseau* masque à l'aide de l'option **netmask**. Enfin, **interface** est l'interface réseau vers laquelle doivent être envoyés les paquets qui vérifient les critères de sélection de cette règle

# GESTION DES HOTES

## ❑ Définition des règles de routage

Par exemple, la règle de routage à utiliser pour l'interface loopback est la suivante :

```
route add -net 127.0.0.0 netmask 255.0.0.0 lo
```

Cette règle signifie que tous les paquets dont l'adresse de destination appartient au sous-réseau de classe A 127.0.0.0 doivent être transférés vers l'interface loopback. Cela implique en particulier que les paquets à destination de la machine d'adresse IP **127.0.0.1** (c'est-à-dire la machine locale) seront envoyés vers l'interface loopback (ils reviendront donc sur la machine locale).

# GESTION DES HOTES

## ❑ Définition des règles de routage

Une autre règle de routage classique est la suivante :

```
route add -net 192.168.0.0 netmask 255.255.255.0 eth0
```

Elle permet d'envoyer tous les paquets à destination du réseau de **classe C 192.168.0.0** vers la première interface Ethernet. C'est typiquement ce genre de règle qu'il faut utiliser pour faire fonctionner un réseau local.

# GESTION DES HOTES

## ❑ Définition des règles de routage

Il n'est normalement pas nécessaire d'ajouter les règles de routage pour les réseaux auquel la machine est connectée. En effet, la configuration d'une carte réseau à l'aide de la commande **ifconfig** ajoute automatiquement à la table de routage une règle pour envoyer les paquets à destination de ce réseau par l'interface réseau qui y est connectée. Cependant, la commande **route** devient réellement nécessaire lorsqu'il faut définir les passerelles à utiliser pour l'envoi des paquets destinés à une machine à laquelle la machine locale ne peut accéder directement. Les règles de routage faisant intervenir une passerelle sont semblables aux règles de routage vues ci-dessus, à ceci près que l'adresse IP de la passerelle à utiliser doit être fournie.



# GESTION DES HOTES

## ❑ Définition des règles de routage

Pour cela, on utilise l'option **gw** (abréviation de l'anglais « Gateway »). La syntaxe utilisée est donc la suivante :

*route add [-net | -host] adresse netmask masque gw passerelle interface*

où passerelle est l'adresse **IP de la passerelle** à utiliser pour router les paquets qui vérifient les critères de cette règle. Les autres paramètres sont les mêmes que pour les règles de routage classique.

# GESTION DES HOTES

## ❑ Définition des règles de routage

Par exemple, supposons qu'une machine soit connectée à un réseau d'adresse **192.168.0.0**, et que sur ce réseau se trouve une passerelle d'adresse **192.168.0.1** permettant d'atteindre un autre réseau, dont l'adresse est **192.168.1.0**. Une machine du réseau **192.168.0.0** aura typiquement les règles de routage suivantes :

```
route add -net 192.168.0.0 netmask 255.255.255.0 eth0
```

```
route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.0.1 eth0
```

La première règle permet, comme on l'a déjà vu, de communiquer avec toutes les machines du réseau local. La deuxième règle permet d'envoyer à la passerelle **192.168.0.1** tous les paquets à destination du réseau **192.168.1.0**.

# GESTION DES HOTES

## ❑ Définition des règles de routage

Inversement, si la passerelle utilise l'adresse **192.168.1.15** sur le réseau **192.168.1.0**, les machines de ce réseau qui voudront accéder au réseau **192.168.0.0** devront spécifier la règle de routage suivante :

***route add -net 192.168.0.0 netmask 255.255.255.0 gw 192.168.1.15 eth0***

Bien entendu, il est nécessaire que toutes les machines des deux réseaux utilisent ces règles de routage pour que la communication entre les deux réseaux se fasse dans les deux sens.

# GESTION DES HOTES

## ❑ Définition des règles de routage

Le problème de ces règles de routage est qu'elles spécifient l'adresse du réseau destination. Il est évidemment hors de question d'utiliser une règle de routage différente pour toutes les adresses de réseaux possibles. Il est donc possible de définir ce qu'on appelle une passerelle par défaut, qui n'est rien d'autre que la passerelle vers laquelle doivent être envoyés tous les paquets qui n'ont pas vérifié les critères des autres règles de routage. La syntaxe à utiliser pour définir la passerelle par défaut est plus simple, puisqu'il n'est plus nécessaire de préciser les critères de sélection :

***route add default gw passerelle interface***

où la signification des paramètres **passerelle** et **interface** est inchangée.

# GESTION DES HOTES

## ❑ Définition des règles de routage

Ainsi, pour reprendre l'exemple précédent, supposons que la machine **192.168.0.47** dispose d'une connexion à Internet et soit configurée pour partager cette connexion avec les machines du réseau local. Pour que toutes les machines du réseau local puisse profiter de cette connexion, il suffit de demander à ce que tous les paquets qui ne vérifient aucune autre règle de routage soient envoyés à la passerelle **192.168.0.47**. Cela se fait avec la règle de routage suivante :

```
route add default gw 192.168.0.47 eth0
```

# GESTION DES HOTES

## ❑ Activer le routage

Par défaut, si une machine Linux a plusieurs interfaces, elle ne sait pas faire office de routeur.

Nous allons mettre en oeuvre l'IP Forwarding. Cela fonctionne quelque soit la distribution Linux.

### ○ Vérifier l'état du routage

Pour vérifier si le système sait router les paquets, on pourra consulter le résultat de :

```
cat /proc/sys/net/ipv4/ip_forward
```

- ❖ *Si la commande retourne 0, l'IP forwarding (routage) n'est pas activé*
- ❖ *Si la commande retourne 1, l'IP forwarding (routage) est activé*

# GESTION DES HOTES

## ❑ Activer le routage

### ✓ Activer / Désactiver le routage

Comme vu précédemment, 0 = désactivé, 1 = activé. Il faudra donc modifier cette variable pour activer ou désactiver le routage.

#### ➤ De manière provisoire

Pour activer provisoirement (non persistant après un reboot) :

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

On peut aussi utiliser la commande sysctl :

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

*On mettra 0 pour désactiver de manière temporaire si c'est activé!*

# GESTION DES HOTES

## ❑ Activer le routage

### ✓ Activer / Désactiver le routage

Comme vu précédemment, **0** = désactivé, **1** = activé. Il faudra donc modifier cette variable pour activer ou désactiver le routage.

### ➤ De manière définitive

Pour rendre le routage définitif (par défaut, désactivé), on va éditer le fichier `sysctl.conf`. Aujourd'hui, ce fichier n'est plus utilisé, on positionnera un fichier, nommé **routeur.conf** dans **/etc/sysctl.d** :

```
vi /etc/sysctl.d/routeur.conf
```

On positionnera cette ligne :

```
net.ipv4.ip_forward=1
```

```
sysctl -p /etc/sysctl.d/routeur.conf (pour prendre en compte le modifications)
```