université de BORDEAUX

TP6 - Routage & Firewall

1 Routage

1.1 Préliminaires

Pour dialoguer avec la Terre entière, il est nécessaire de passer par des passerelles, appelées en anglais *gateway*. Lancez la commande `/sbin/route -n` . Il y a deux routes importantes.

- La route des machines de la salle (en `10.0.x.y`), qui sont accessibles directement en Ethernet sans passer par un routeur, remarquez le `Genmask`, il indique la séparation entre la partie réseau (bits à 1) et la partie machine des adresses IP (bits à 0). Comparez avec l'adresse d'une machine voisine dans la même salle : êtes-vous bien dans le même réseau ?
- La route par défaut (`0.0.0.0`), qui utilise une passerelle (quelle est son adresse IP ?)

On peut aussi utiliser la version plus moderne `ip route ls` . À quoi correspond le suffixe `/24` ?

Pour observer en IPv6, on utilise `ip -6 route ls` . On a les mêmes deux routes importantes :

- La route des machines de la salle (en `2001:660:6101:800:x::y`), qui sont accessibles directement en Ethernet. Observez le suffixe `/80`, et comptez le nombre de chiffres hexadécimaux auxquels cela correspond dans l'adresse IP pour déterminer la partie réseau et la partie machine. (attention, les 0 non significatifs ne sont pas écrits en IPv6, chaque paquet séparé par `:` compte pour 16 bits).
- La route des adresses locales `fe80::/64` : on a sur `eth0` non seulement une adresse publique (`2001:...`), mais aussi une adresse locale en `fe80::/64`. L'une ou l'autre sera utilisée selon que l'on veut émettre en local seulement, ou bien sur le reste d'Internet.
- La route par défaut (`default`) utilise une passerelle de type `fe80::...` : c'est l'adresse IPv6 locale du routeur pour cette salle-ci.

Memento Routage

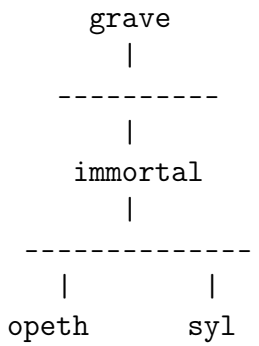
En guise de documentation rapide, voici un résumé des commandes que l'on va utiliser dans les sections suivantes. Il n'y a donc pas d'action à faire dans cette section, c'est juste de la documentation. Il faut bien sûr remplacer `<@gateway>` par une adresse IP, etc.

- Activer le relai des paquets sur une machine (ip forward) :
  `echo 1 > /proc/sys/net/ipv4/ip_forward`

— Configurer de manière permanente le relai des paquets : voir le fichier `/etc/sysctl.conf`
— Afficher la table de routage : `route -n`
— Définir une route par défaut : `route add default gw <@gateway>`
— Ajouter une route vers un réseau spécifique :
`route add -net <@network/bits> gw <@gateway>`
— Pour supprimer une règle, il faut taper la commande `route del <...>` avec exactement les mêmes arguments que pour la commande `add`.
— Vérifier la route envoyer un paquet à une adresse spécifique : `traceroute <@host>`
— On peut aussi utiliser la commande `ip` pour ajouter des route :
`ip route add <@network/bits> via <@gateway>` (de même pour une seule adresse).

1.2 Routage Basique

Dans un LAN, toutes les machines peuvent communiquer directement, car elles sont physiquement connectées par leurs interfaces réseaux. Dans un réseau plus complexe, comme celui que nous allons étudier maintenant, il est nécessaire de configurer les tables de routage des machines, pour qu'elles collaborent à l'acheminement des messages d'un bout à l'autre du réseau.



Pour lancer la topologie ci-dessus, veuillez taper la commande suivante sur votre machine au CREMI :

```
$ /net/ens/qemunet/qemunet.sh -x -s /net/ens/qemunet/demo/gw1.tgz
```

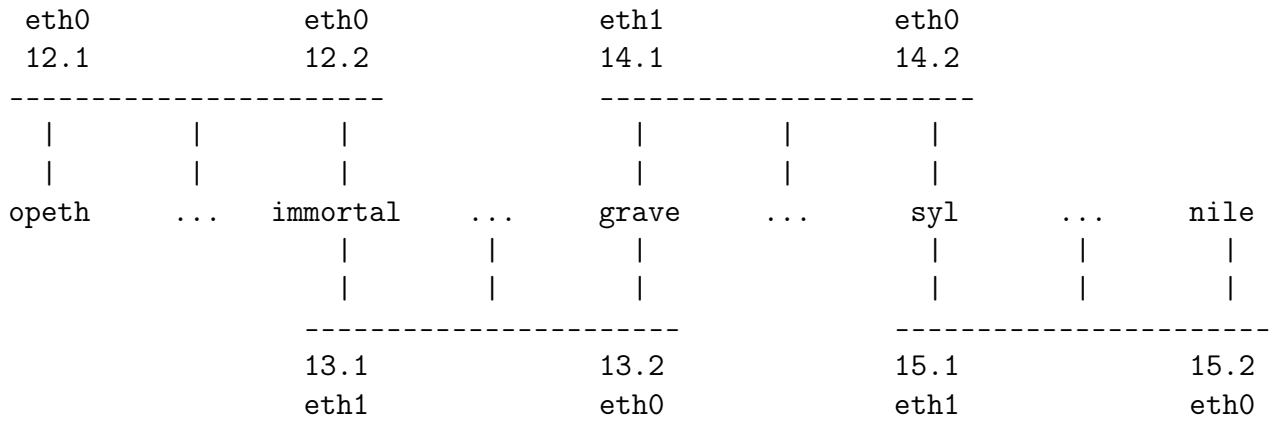
Si cela ne se lance pas, vérifiez que vous n'avez pas virtualbox lancé (qui vient en conflit pour la virtualisation, il faut donc le fermer), et que vous le lancez bien directement sur votre machine.

— Les adresses IPs sont déjà configurées. Veuillez reporter chaque adresse IP sur le schéma ci-dessus en précisant à chaque fois le nom de l'interface ethX. Pour vous aider, vous pouvez consulter, sur la machine hôte, le fichier `demo/gw.topo` ce fichier détaille la configuration du réseau virtuel dans QemuNet. Puisque certaines machines ont plusieurs adresses IP, il vaut mieux systématiquement donner aux commandes des adresses IP, et non des noms de machines.
— Vérifiez avec 'ping' que les machines peuvent communiquer dans leurs réseaux locaux respectifs.
— Afficher les tables de routage avec la commande 'route -n'.

— Configurez les tables de routage des différentes machines à l'aide de la commande 'route', pour que tout le monde puisse communiquer avec tout le monde. Il faut également activer le relai des paquets sur `immortal` pour qu'il agisse comme un routeur. On rappelle quelques éléments de syntaxe dans le memento ci-dessous. Vérifiez avec ping que toutes les machines sont capables de communiquer ensemble. Si ça ne passe pas, utilisez `tcpdump -n -i any` sur les différentes machines pour voir où ça coince.
— Faites un ping entre opeth et grave. Lancez `tcpdump -n -i any` sur immortal afin d'afficher le trafic qui circule...
— Si vous ne l'avez pas fait, simplifiez le routage en utilisant `default` plutôt que des routes explicites.

1.3 Routage Avancé

Voici une nouvelle configuration, composée de 4 sous-réseaux /24 dans le réseau 147.210.0.0/16 :



Démarrez ce réseau virtuel :

```
$ /net/ens/qemunet/qemunet.sh -x -s /net/ens/qemunet/demo/chain0.tgz
```

Les adresses IP sont déjà configurées. Il faut donc configurer les tables de routage afin que tout le monde puisse communiquer avec tout le monde. La machine *grave* nécessite l'utilisation de routes spécifiques (option `-net`), les autres peuvent utiliser des routes `default`

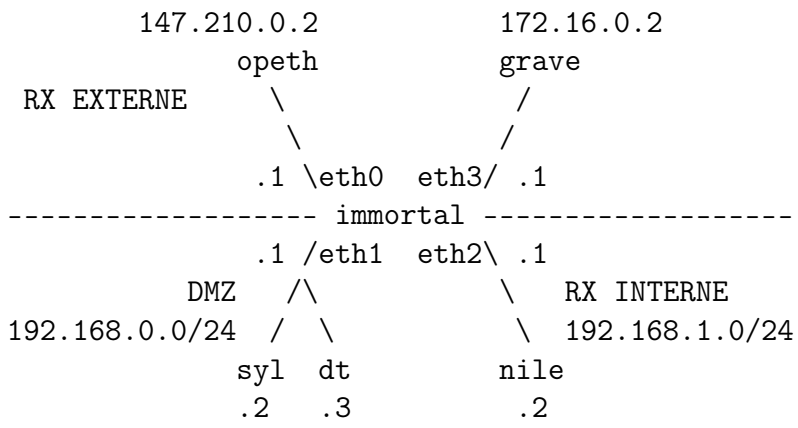
2 Firewall

Lancez la configuration *QemuNet* suivante[1] :

```
$ /net/ens/qemunet/qemunet.sh -x -s /net/ens/qemunet/demo/dmz.tgz
```

1. Adaptez la commande si vous souhaitez lancer *QemuNet* à distance avec *tmux*.

Note : si vous avez à un moment un problème d'affichage (parce que vous avez retaillé la fenêtre), utilisez la commande `resize` pour prévenir la VM que la fenêtre a changé de taille.
Voici la topologie de notre réseau :



Les IP et les tables de routage sont déjà configurées pour vous :-) Par défaut, il n'y a aucun firewall en place, ce qui signifie que tout le monde peut communiquer avec tout le monde sans restriction particulière.

Dans cet exercice, nous jouons le rôle de l'administrateur d'un petit réseau d'entreprise, rélié à Internet par la passerelle *immortal*. "Internet" est ici représenté par seulement les deux machines `opeth` est `grave`. Nous allons configurer un firewall sur cette passerelle à l'aide de la commande `iptables`. Un memento dans la section suivante vous donne la syntaxe de cette commande pour vous aider, commencez par y jeter un coup d'oeil. D'autre part, rapidement vous aurez besoin d'utiliser `tcpdump` pour vérifier à quel endroit quels paquets parviennent à passer ou restent au contraire bloqués, utilisez-le vraiment abondamment !

À noter que `iptables` est en train d'être supplanté par `nftables`, mais dans un futur proche, c'est encore `iptables` que vous verrez en production, et les principes sont les mêmes de toutes façons.

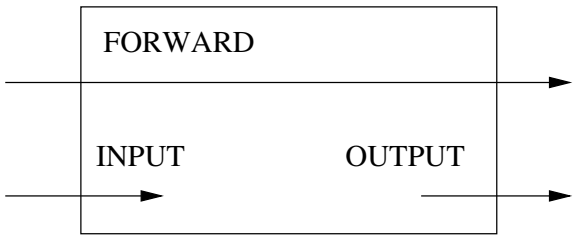
— Au sein d'un réseau d'entreprise, quel différence y a-t-il entre la DMZ et le réseau interne des employés ?
— Sur *immortal*, positionnez la politique par défaut à DROP :
```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```
— Nous venons donc d'activer le firewall sur *immortal*. Plus aucun trafic réseau n'est autorisé vers ou à travers *immortal*. Vérifiez avec ping.

Nous allons maintenant ajouter des règles pour autoriser explicitement seulement certains trafics réseau. Toutes les questions suivantes nécessitent donc de taper des commandes (sur *immortal* uniquement) de la forme : `iptables -A FORWARD ... -j ACCEPT` , cf le memento plus bas pour les détails de syntaxe. Si vous avez fait une erreur, vous pouvez supprimer une ligne en reprenant la même commande et en remplaçant `-A` par `-D`

— Autorisez le ping (c'est-à-dire le protocole icmp) du réseau Interne vers Internet, sans autoriser l'inverse.
— Faites un test *ping*, constatez que cela ne fonctionne pas : il faut effectivement autoriser la réponse à passer ! Pour simplifier l'autorisation des réponses de manière générale, on peut utiliser
`iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT`
qui laissera passer tout ce qui est identifié comme réponse à ce que l'on a déjà laissé passer auparavant (c'est donc un "catch-all" pour les réponses).
— Autorisez l'accès à tous les serveurs web depuis les machines du réseau interne. Faites un test avec *wget* vers opeth et vers grave.
— Autorisez *grave* à accèder au serveur ssh (port 22) de *dt*. Faites un test avec le compte `toto` (mot de passe `toto`).
— Autorisez l'accès depuis n'importe où vers le serveur web de syl (port 80). Faites un test avec wget.
— Depuis opeth et grave, testez votre firewall sur les machines de votre DMZ avec nmap ! (utilisez l'option `-PN` puisque l'on n'a pas ouvert le ping)

Memento Firewall

La configuration du firewall se base sur la table "filter" de la commande `iptables`. Elle est subdivisée en 3 chaînes, notée <CHAIN> :
— INPUT : tout ce qui est à destination de la machine elle-même ; cela ne concerne donc pas les paquets qui seront relayés
— OUTPUT : tout ce qui est émis par la machine elle-même ; cela ne concerne donc pas les paquets qui seront relayés
— FORWARD : tout ce qui ne fait que traverser la machine (donc les paquets relayés).



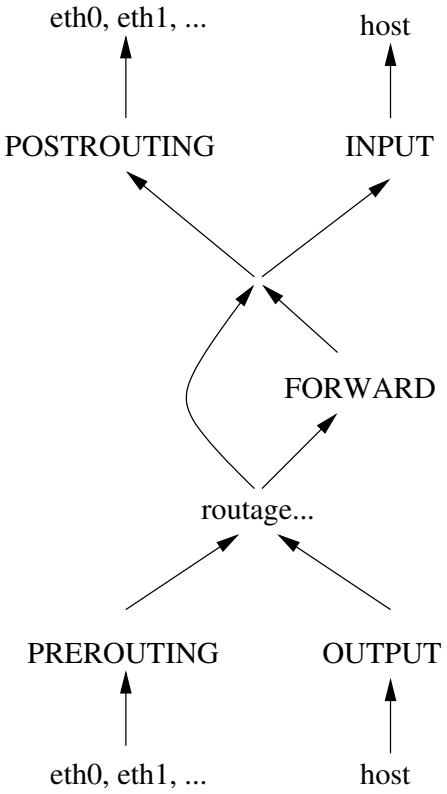
Voici la syntaxe de principales commandes `iptables` :
— Pour afficher les règles de la table filter : `iptables -L -v`
— Pour effacer toutes les règles ajoutées : `iptables -F`
— Pour chaque règle que l'on ajoute, trois actions sont possibles (notée <ACTION>) :
  — ACCEPT : on accepte ;
  — REJECT : on rejette poliment (réponse d'erreur envoyé à l'émetteur) ;
  — DROP : on jette à la poubelle (pas de réponse d'erreur).
— Pour modifier la politique par défaut du firewall :
`iptables -P <CHAIN> <ACTION>`
— Pour ajouter une nouvelle règle à une chaîne du firewall :
`iptables -A <CHAIN> <SRC> <DST> <...> -j <ACTION>`

  - avec <SRC> des indications éventuelles sur la provenance des paquets IP, comme par exemple `-i eth0` ou `-s 192.168.0.0/24` ;
  - avec <DST> des indications éventuelles sur la destination des paquets IP, comme par exemple : `-o eth1` ou `-d 147.210.0.0/24` ;
  - avec <...> des infos complémentaires par exemple sur la nature du protocole `-p icmp` , `-p tcp` , ou `-p udp` avec après éventuellement des précisions spécifiques à ces protocoles (`--dport 80` pour TCP) ou encore sur l'état `-m state --state NEW`, ...

  - sur l'état, il est notamment utile d'employer `-m state --state RELATED, ESTABLISHED` pour identifier les paquets qui sont une réponse à quelque chose que l'on a déjà laissé passer : une réponse à un ping, une réponse à une demande de connexion TCP, etc.
- Pour supprimer une règle, on peut utiliser `iptables -D <CHAIN>` en mettant derrière soit un numéro de règle (lisible avec `iptables -L -v --line-numbers`, soit la règle elle-même (i.e. remplacer `-A` par `-D`).
- Note : le principe est que c'est la première règle qui concorde qui l'emporte. Ainsi, il faut que la liste commence par les cas les plus précis avant de traiter les cas les plus généraux.
- Pour ajouter une règle au début de la liste, on peut utiliser `iptables -I <CHAIN>`
- Pour supprimer toutes les règles, on peut utiliser `iptables -F <CHAIN>`

Pour plus d'info, consulter le manuel : `man iptables` et `man iptables-extensions`

Pour les plus avancés, vous pouvez également consulter le manuel à propos des tables `nat` et `mangle` (qui permettent de bricoler les paquets, notamment les adresses source et destination) et les chaînes `PREROUTING` et `POSTROUTING`. Le principe est ainsi :



Selon que le paquet arrive depuis une carte réseau ou vient du firewall lui-même, on passe par `PREROUTING` ou par `OUTPUT`. On passe alors l'étape de routage. Si notre paquet est arrivé depuis une carte réseau et que d'après le routage il doit repartir sur une carte réseau, on passe par `FORWARD`. Enfin, selon que le paquet doit repartir sur une carte réseau ou est à destination du firewall lui-même, on passe par `POSTROUTING` ou par `INPUT`.

On peut ainsi par exemple utiliser `PREROUTING` pour changer l'adresse IP destination des connexions venant d'Internet vers le port 80, pour l'envoyer vers un serveur web du réseau local. Ou inversement, utiliser `POSTROUTING` pour changer l'adresse IP source des connexions venant du réseau local vers Internet (`MASQUERADE`).