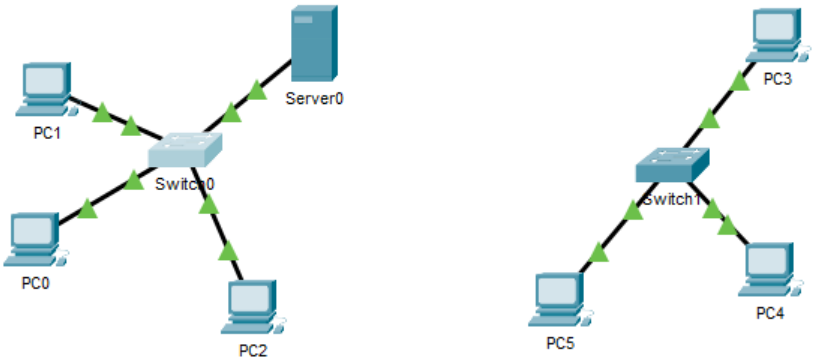
TP cyber sécurité 2

Exercice 1: Début du routage

En utilisant Packet tracer sur TSE1 via guacamole, refaites le schéma suivant :



Nous avons 2 réseaux différents. Choisissez un réseau de classe C pour le réseau 1 et un réseau de classe B pour le réseau 2, et mettez les adresses IP sur les Pcs.

Faites un ping entre les différents Pcs (Onglet Desktop, terminal prompt).

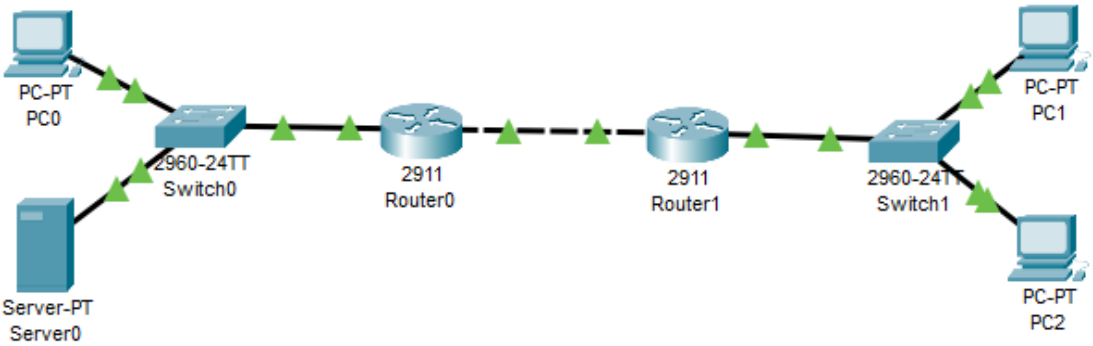
Nous avons 2 réseaux distincts, on rajoute un routeur (ex :2911) entre les deux switches. Un routeur fait automatiquement le lien entre deux réseaux qui lui sont connectés.

Regardez la table de routage.

Ajoutez une passerelle de sortie aux différents PCs (default gateway). Refaites des pings, et regardez ce que cela donne.

Exercice 2: le ROUTAGE et le NAT

En utilisant Packet tracer sous linux, refaites le schéma suivant :



Les routeurs sont des routeurs 2911.

1) Combien y-a-t-il de réseaux différents sur ce schéma ?
2) Faites un plan d'adressage IP et utilisez le dans Packet tracer.

3) Une fois que toutes les interfaces ont un point vert (cf. schéma), vérifiez que le PC0 peut pinger le server0, et que le PC1 peut pinger le PC2. (Onglet Desktop, terminal prompt).
4) Maintenant, en mode temps réel, faites un ping entre le PC0 et le PC1. Est-ce que cela fonctionne ? si oui, tant mieux, si non...
    - Visualisez la table de routage, elle ne permet pas d'atteindre le réseau distant.
    - Dans l'onglet, config-> routing-> static, rajouter la ligne manquante.

5) Une fois que tout fonctionne, repassez en mode simulation. Faites un ping entre le PC0 et PC1. Dans le "simulation panel", des trames apparaissent. Cliquer sur le carré de couleur d'une trame pour voir son contenu. Observez les informations de la couche 2 et de la couche 3 (@MAC, @IP),
6) Maintenant, on installe du PAT sur le Router 0
    - Pour simplifier le TP, télécharger directement le fichier pkt à l'adresse : https://perso.isima.fr/~palauren/packet/nat.pkt
    - Faites la même chose que la question 5.
    - Sur le CLI du routeur0, essayer la commande : #show ip nat translation
      Qu'observez-vous ?

Exercice 3: Visualisation des ports réseaux

1) Pour connaître les ports ouverts d'une machine à partir d'elle même
   - utilisation de la commande : netstat ou ss (sous linux)
   - quelles options utilisées ? (cf man ou help)

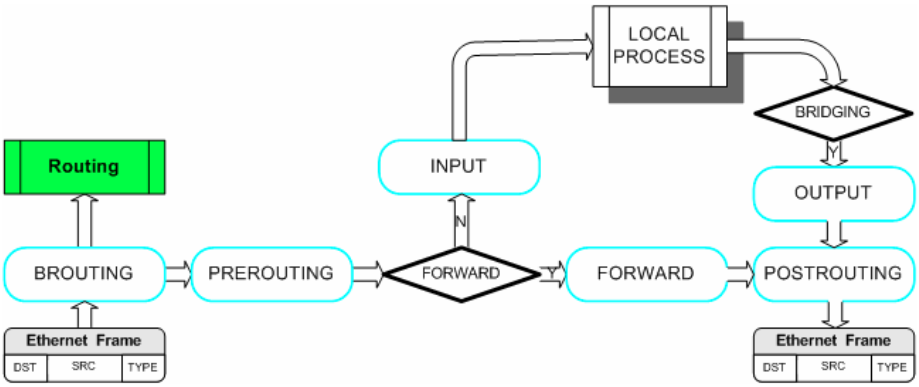
2) Connaître les ports ouverts d'une machine à partir d'une autre machine
   - commande nmap -sS ou -sU ou -A @IP

3) Sur un PC, lancez le serveur http : *sudo apachectl start*
   a. *Vérifiez avec nmap et netstat que le serveur http tourne*
   b. *Arrêter apache : apachectl stop*

Exercice 4: Pare-feu sous linux

Sous linux, il existe un utilitaire qui sert de pare-feu : iptables. Ce programme sert à manipuler les règles de filtrage de paquets au niveau du noyau Linux. Le noyau dispose de listes de règles appelées **chaînes**. Les règles sont analysées les unes à la suite des autres dans l'ordre de leur écriture. Les chaînes sont regroupées dans plusieurs tables dont :

- la table NAT (permet de faire du NAT/PAT), avec les chaînes suivantes :
    - PREROUTING
    - POSTROUTING
- la table FILTER, avec les chaînes suivantes :
    - INPUT
    - OUTPUT
    - FORWARD



Un paquet rentrera toujours dans la machine via la chaîne PREROUTING et sortira toujours de la machine via la chaîne POSTROUTING. Si le paquet doit être routé, il passera dans la chaîne FORWARD. Les chaînes INPUT et OUTPUT quant à elles serviront respectivement à placer des règles pour les paquets destinés au et émis par le pare-feu lui-même.
Chaque chaîne peut fonctionner selon trois politiques différentes :

ACCEPT: tous les paquets sont acceptés.
DROP: les paquets sont refusés sans notification à l'émetteur des paquets.
REJECT: les paquets sont refusés mais avec notification à l'émetteur des paquets.

A l'aide des règles affectables à chaque chaîne, il est possible d'autoriser, de restreindre ou d'interdire l'accès à différents services réseaux, et ainsi modifier la politique de filtrage des paquets de chaque chaîne.

Les commandes :

`-A --append` : Ajoute la règle à la fin de la chaîne spécifiée

Exemple :`# iptables -A INPUT ...`

`-D --delete` : Permet de supprimer une chaîne. On peut l'utiliser de 2 manières, soit en spécifiant le numéro de la chaîne a supprimer, soit en spécifiant la règle à retirer.

```
# iptables -D INPUT --dport 80 -j DROP
# iptables -D INPUT 1
```

`-L --list` : Permet d'afficher les règles.

`# iptables -L` # Affiche toutes les règles des chaînes de FILTER

`# iptables -L INPUT` # Affiche toutes les règles de INPUT (FILTER)

`-F --flush` : Permet de vider toutes les règles d'une chaîne.

Exemple :`# iptables -F INPUT`

`-N --new-chain` : Permet de créer une nouvelle chaîne.

Exemple :`# iptables -N LOG_DROP`

`-X --delete-chain` : Permet d'effacer une chaîne.

Exemple :`# iptables -X LOG_DROP`

Les options :

`-j (jump)` : Défini l'action à prendre si un paquet répond aux critères de cette règle: `ACCEPT`, `LOG`, `DROP`...

Exemple :`# iptables -A INPUT -p icmp -j DROP`

`-p --protocol` : Spécifier un protocole : `tcp`, `udp`, `icmp`, `all` (tous)

Exemple :`# iptables -A INPUT -p icmp -j DROP`

`-s --source` : Spécifier une adresse source à matcher

Exemple :`# iptables -A INPUT -p tcp -s 192.168.42.42 -j ACCEPT`

`-d --destination` : Spécifier une adresse destination
Exemple :`# iptables -A FORWARD -p tcp -d 10.1.0.1 -j ACCEPT`
`-i --in-interface` : Spécifier une interface d'entrée
Exemple :`# iptables -A INPUT -p icmp -i eth0 -j DROP`
etc... voir man iptables

Cet exercice se fait avec un binôme pour tester votre configuration

1) Mettre en place une règle de filtrage pour interdire le ping

2) Utilisez des règles de filtrage pour que seul votre binôme puisse accéder en ssh à votre ordinateur, mais personne d'autres.