Configuration de l'infrastructure réseau virtuelle sous VMware

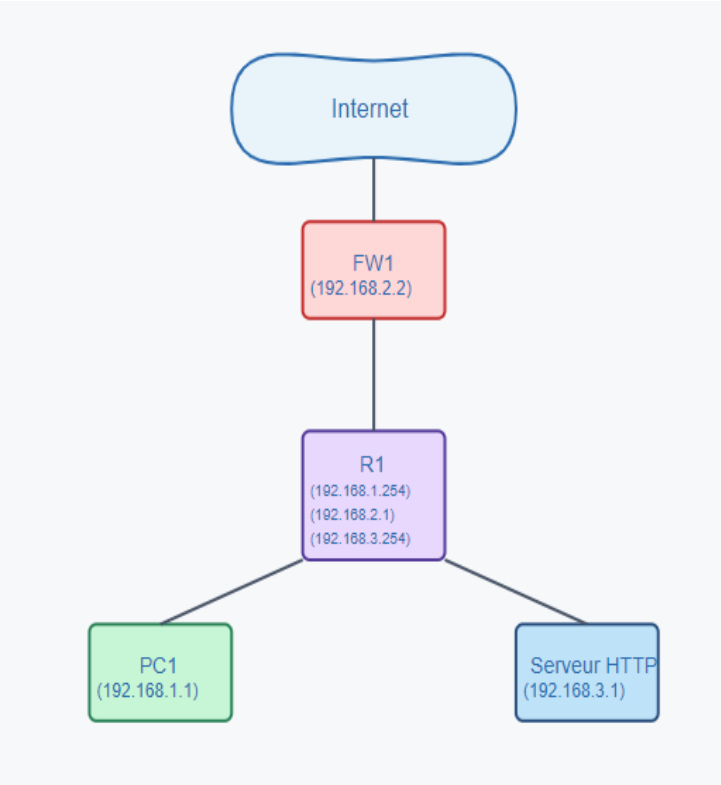
Étape 1 : Création de la maquette dans VMware

Pour créer mon environnement de test, je commence par cloner une machine que j'avais auparavant (une Kali Linux) sous VMware Workstation. Je procède comme suit :

Noms des VMs :

a. **PC1** (poste de travail)
b. **FW1** (pare-feu)
c. **R1** (routeur)
d. Serveur HTTP

Ces machines seront connectées entre elles selon la topologie réseau suivante :



Configuration des réseaux virtuels

Pour structurer les sous-réseaux, j'ajoute des réseaux virtuels sous VMware en suivant les étapes ci-dessous :

1. *Edit → Virtual Network Editor → Add Network*
    a. **vmnet2** : pour le réseau entre **PC1** et **R1** - 192.168.1.0/24
    b. **vmnet3** : pour le réseau entre **R1** et **FW1** - 192.168.2.0/24
    c. **vmnet4** : pour le réseau entre **R1** et **Serveur HTTP** - 192.168.3.0/24
2. Je désactive le service DHCP pour chacun de ces réseaux, car je vais configurer manuellement les adresses IP.

Configuration des interfaces réseau des VMs

Ensuite, je configure les adaptateurs réseau de chaque VM en fonction du sous-réseau approprié :

- **PC1**
    - *Network Adapter → Custom*: **vmnet2**
- **FW1**
    - *Network Adapter 1* → NAT (accès Internet)
    - *Add Network Adapter → Custom*: **vmnet3**
- **R1**
    - *Network Adapter 1 → Custom*: **vmnet2**
    - *Add Network Adapter → Custom*: **vmnet3**
    - *Add Network Adapter → Custom*: **vmnet4**
- **Serveur HTTP**
    - *Network Adapter → Custom*: **vmnet4**

Tableau récapitulatif des interfaces

Machine	Interface	Adresse IP	Passerelle
PC1	eth0	192.168.1.1/24	192.168.1.254
FW1	eth1	192.168.2.2/24	-
	eth0	DHCP	-
R1	eth0	192.168.1.254/24	-
	eth1	192.168.2.1/24	-
	eth2	192.168.3.254/24	-
Serveur HTTP	eth0	192.168.3.1/24	192.168.3.254

Configuration des adresses IP sur les interfaces

Dans chaque VM, j'assigne les adresses IP respectives :

- **PC1** :
  - `sudo ip addr add 192.168.1.1/24 dev eth0`
- **FW1** :
  - `sudo ip addr add 192.168.2.2/24 dev eth1`
- **R1** :
  - `sudo ip addr add 192.168.1.254/24 dev eth0`
  - `sudo ip addr add 192.168.2.1/24 dev eth1`
  - `sudo ip addr add 192.168.3.254/24 dev eth2`
- **Serveur HTTP** :
  - `sudo ip addr add 192.168.3.1/24 dev eth0`

Étape 2 : Configuration du routage

Activation du routage

Pour permettre le routage entre les sous-réseaux, j'active l'option de routage IPv4 sur **R1** et **FW1** :

- `sudo sysctl -w net.ipv4.ip_forward=1`
- `echo "net.ipv4.ip_forward=1" | sudo tee /etc/sysctl.d/99-routing.conf`
- `sudo sysctl -p /etc/sysctl.d/99-routing.conf`

Configuration des routes

- **PC1** :
  - `sudo ip route add default via 192.168.1.254`
- **Serveur HTTP** :
  - `sudo ip route add default via 192.168.3.254`
- **R1** :
  - `sudo ip route add 192.168.1.0/24 dev eth0`
  - `sudo ip route add 192.168.2.0/24 dev eth1`
  - `sudo ip route add 192.168.3.0/24 dev eth2`
  - `# Default Route`
  - `sudo ip route add default via 192.168.2.2`
- **FW1** :
  - `sudo ip route add 192.168.1.0/24 via 192.168.2.1`
  - `sudo ip route add 192.168.3.0/24 via 192.168.2.1`

Étape 3 : Configuration du NAT et du Pare-feu

Configuration de NAT et des règles de filtrage

J'installe et configure **iptables** sur **FW1** pour activer le NAT et rediriger le trafic vers le serveur HTTP :

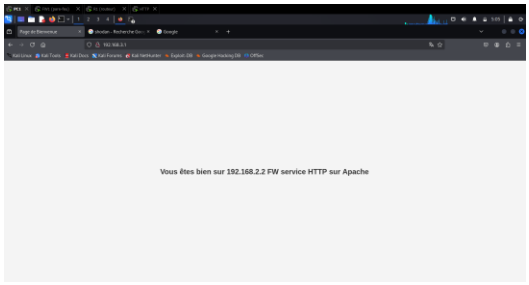
```
#installation iptables
sudo apt update
sudo apt install iptables iptables-persistent


#>>>>SCRIPT IPTABLES BLOCKED ICMP
#nettoyage/flush les rules existantes
sudo iptables -t nat -F
sudo iptables -F
#politiques par défaut
sudo iptables -P FORWARD DROP
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
#bloquer ICMP (ping)
sudo iptables -A FORWARD -p icmp -j DROP  # Bloque tous les ICMP qui traversent le pare-feu
sudo iptables -A INPUT -p icmp -j DROP    # Bloque les pings vers le pare-feu lui-même
#rules forwarding
sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
#allow forwarding depuis tous les réseaux internes vers internet
sudo iptables -A FORWARD -i eth1 -o eth0 -p tcp -j ACCEPT  # TCP
sudo iptables -A FORWARD -i eth1 -o eth0 -p udp -j ACCEPT  # UDP
sudo iptables -A FORWARD -s 192.168.3.0/24 -o eth0 -p tcp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.3.0/24 -o eth0 -p udp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.1.0/24 -o eth0 -p tcp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.1.0/24 -o eth0 -p udp -j ACCEPT
#configuration NAT
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
#sauvegarder les rules et les lister
sudo netfilter-persistent save
sudo netfilter-persistent reload
sudo iptables -L
```

Vérifications et Tests de Connectivité

Je vérifie la connectivité entre les machines pour m'assurer que tout fonctionne correctement :

Je m'assure de débloquer le protocol ICMP avant de faire les tests suivats :

```
#>>>>SCRIPT IPTABLES ALLOWED ICMP
#nettoyage/flush les rules existantes
sudo iptables -t nat -F
sudo iptables -F
#politiques par défaut
sudo iptables -P FORWARD DROP
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
#rules forwarding
sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
#allow ICMP (ping)
sudo iptables -A FORWARD -p icmp -j ACCEPT
sudo iptables -A INPUT -p icmp -j ACCEPT
#allow forwarding vers internet depuis sous-réseaux
sudo iptables -A FORWARD -i eth1 -o eth0 -p tcp -j ACCEPT
sudo iptables -A FORWARD -i eth1 -o eth0 -p udp -j ACCEPT
sudo iptables -A FORWARD -i eth1 -o eth0 -p icmp -j ACCEPT # ICMP
sudo iptables -A FORWARD -s 192.168.3.0/24 -o eth0 -p tcp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.3.0/24 -o eth0 -p udp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.3.0/24 -o eth0 -p icmp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.1.0/24 -o eth0 -p tcp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.1.0/24 -o eth0 -p udp -j ACCEPT
sudo iptables -A FORWARD -s 192.168.1.0/24 -o eth0 -p icmp -j ACCEPT
#configuration nat
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
#sauvegarder et lister les rules
sudo netfilter-persistent save
sudo netfilter-persistent reload
sudo iptables -L
```

- **Tests de ping** :
  - **Depuis PC1** :
    - `ping 192.168.1.254`
    - `ping 192.168.3.1`
    - `ping 192.168.2.2`
  - **Depuis R1** :
    - `ping 192.168.1.1`
    - `ping 192.168.3.1`
    - `ping 192.168.2.2`
- **Tests d'accès Internet** :
  - **Depuis PC1** :
    - `ping 8.8.8.8`
    - `traceroute 8.8.8.8`
- **Test du Serveur HTTP** :
  - Installation d'Apache sur le Serveur HTTP :
    - `sudo apt install apache2`
  - Depuis PC1, je tente d'accéder au Serveur :
    - 

Récapitulatif

Cette config permet :

- La communication entre tous les réseaux internes
- L'accès à Internet depuis tous les réseaux internes via le NAT
- Le routage correct entre tous les segments du réseau
- ICMP (ping) est bloqué
- TCP et UDP sont autorisés pour l'accès Internet
- Le NAT fonctionne toujours pour tous les réseaux internes

Commandes Utiles

1. **État des interfaces** : ip addr show
2. **Tables de routage** : ip route show
3. **Règles NAT** : sudo iptables -t nat -L -v -n
4. **Logs système** : sudo tail -f /var/log/syslog
5. `wget google.com`
6. `ping`
7. `curl`
8. `cewl http://192.168.3.1`