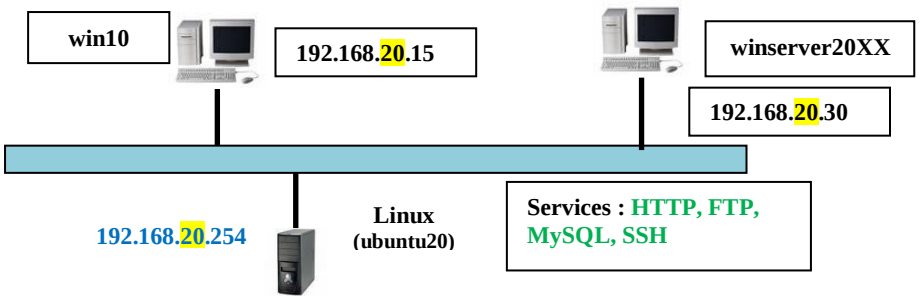
# TP-3 Architecture Systèmes DEUST IOSI 2

Le 02/12/2020 H. TSOUNGUI

## Mise en œuvre du filtrage TCP (pare-feu) sur un serveur linux

-Architecture réseau

-Réseau à configurer : **192.168.20.0**/255.255.255.0




Objectif

L'objectif de ce TP est la mise en œuvre du filtrage sur un serveur en réseau TCP/IP. En fait, il s'agit de configurer un pare-feu pour protéger l'accès aux services disponibles sur le serveur.

A)-Configuration préalable : DEUX machines win10 et Linux

Configurer la machine win10 en lui attribuant une adresse IP, un masque et une adresse de passerelle cohérents et tester la communication entre win 10 et Linux.

-Installer si ce n'est pas encore fait, les services indiqués : HTTP, FTP, SSH, MySQL.

-Tester l'accès à ces services et réaliser les captures nécessaires.

Pour tester cette communication, utiliser par exemple des commandes PING entre les deux machines. Cependant, vous pouvez aussi tester cette communication en essayant d'accéder au service SSH de Linux à partir de win10.

B)-Faire de même entre la machine winServ et Linux : installer la machine winServ et tester les accès aux services. IL EST CONSEILLE d'ARRÊTER win10 pendant ce temps.

C)-**Filtrage des paquets TCP au niveau de la machine Linux**

Quels que soient les résultats obtenus dans la partie, il vous est demandé de mettre en œuvre le FILTRAGE des paquets entre les machines win10 et Linux, puis après avoir arrêté win10 si vous n'avez pas assez de mémoire, faire de même entre winServer et linux.

-Appliquez donc les règles de filtrage données dans le **tableau ci-dessous**. Vous devez expressément utiliser les commandes **iptables** pour activer ces règles de filtrage. Iptables-save sauve les règles.

-Créer un fichier pour sauvegarder ces règles de filtrage.

On donne la syntaxe iptables (voir documentation iptables sur http://tsoungui.fr )

- Syntaxe de iptables :

```
iptables [-t TABLE (Filter, NAT, Mangle)]
         -A Chaîne (INPUT, OUTPUT, FORWARD)
         -i interface_source
         -j interface_sortie
         -s IP_source  -d IP_destination
         --dport port_destination
         -p protocole
         -j ACTION (ACCEPT, DROP, REJECT, LOG)
```

Exemple : **#iptables –A INPUT –s localhost –d localhost –p icmp –j DROP** qui interdit de se pinguer soit-même (localhost)

Schéma de représentation du filtrage par un pare-feu

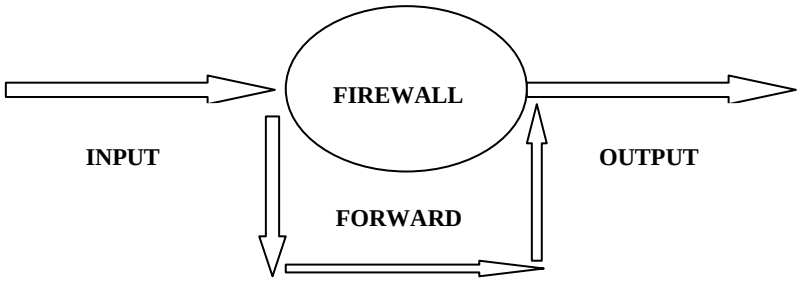



Table des règles de filtrage à mettre en œuvre et tester

| **Destination** | **Source** | **Protocole / Port** | **ACTION sur le paquet** |
|---|---|---|---|
| Linux | Win10 | HTTP / 80 | **Accepter** |
| Linux | Win10 | SSH / 22 | Accepter |
| Linux | Win10 | FTP / 21 | Refuser |
| Linux | Win10 | ICMP (ping) | Refuser |
| | | | |
| Linux | winServ | MySQL / 3306 | Accepter |
| Linux | winServ | HTTP / 80 | Refuser |
| Linux | winServ | SSH / 22 | Accepter |
| * | * | * | Refuser |
| | | | |

Quelques commandes iptables :
-iptables –L liste toutes les règles
-iptables –F supprime toutes les règles actuelles
-iptables-save sauve les règles

NB : les règles sont exécutées séquentiellement de la première à la dernière.
Dès qu'une règle convient, les suivantes ne sont plus exécutées.