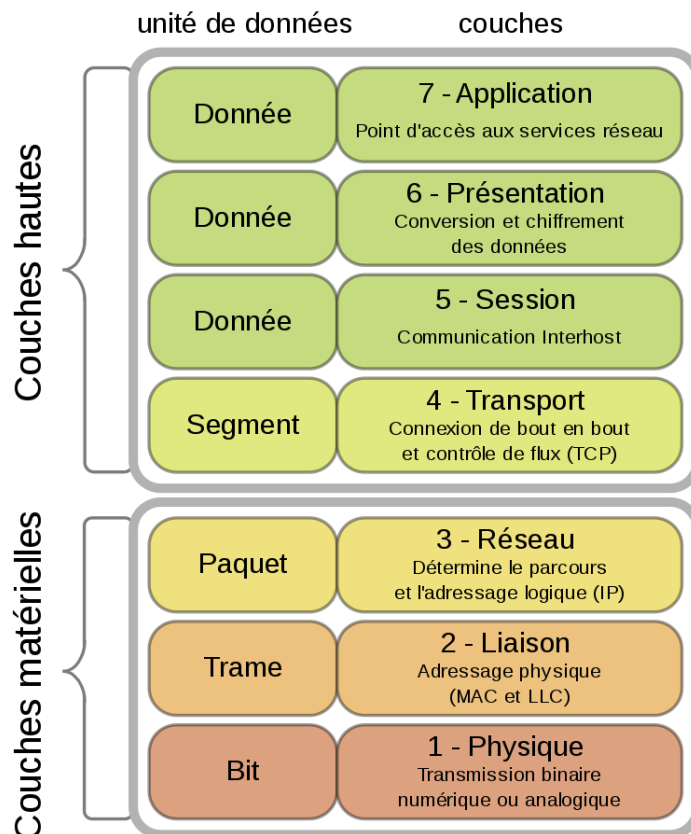


TP IPTABLES



I- Présentation



Netfilter travaille sur les couches 2 (Liaison), 3 (Réseau), 4 (Transport) et 7 (Application).

Présentation :

Depuis la version 2.4, Linux contient un module destiné au filtrage réseau, Netfilter. Il se configure au moyen d'un outil appelé iptables.

Avantages	Inconvénients
• Netfilter est livré avec le noyau 2.4 il est présent dans n'importe quelle machine utilisant ce noyau. • Pas de différenciation entre un simple poste de travail et un serveur de production. • Possède des fonctionnalités de filtrage très avancées. • Commande assez souple	• Pas de moyen simple pour permettre la tolérance de panne ou la répartition de charge de firewall sous Linux. • Netfilter n'offre aucun moyen de mettre en place un cluster dédié au filtrage. • Il n'est pas possible de changer l'état de ces tables dynamiquement.

Tables principales de Netfilter :

Table	Description
filter	Cette table est responsable du filtrage (bloquer ou permettre à un paquet de continuer). Chaque paquet passe à travers cette table, et traverse une et une seule des chaînes prédéfinies suivantes : • Chaîne INPUT : Tous les paquets destinés à cet ordinateur passent dans cette chaîne. • Chaîne OUTPUT : Tous les paquets créés par cet ordinateur passent dans cette chaîne. • Chaîne FORWARD : Tous les paquets traversant cet ordinateur passent dans cette chaîne.
nat	Cette table est responsable de la traduction d'adresses (modification des adresses et/ou des ports des paquets). Le premier paquet de chaque connexion passe à travers cette table. Cette table contient les chaînes prédéfinies suivantes : • Chaîne PREROUTING : Permet la traduction d'adresses de destination uniquement pour les paquets entrant dans l'ordinateur. Le nom de la chaîne vient du fait que les paquets passent cette chaîne avant de passer dans les tables de routage. • Chaîne POSTROUTING : Permet la traduction d'adresses source uniquement pour les paquets sortant de l'ordinateur. Le nom de la chaîne vient du fait que les paquets passent cette chaîne après avoir passé les tables de routage. • Chaîne OUTPUT : Permet la traduction d'adresses de destination uniquement pour les paquets créés par l'ordinateur.
mangle	Cette table est responsable de la transformation des options des paquets, comme la qualité de service. Chaque paquet passe à travers cette table. Comme cette table est prévue pour des options avancées, elle contient toutes les chaînes prédéfinies : • Chaîne PREROUTING : Tous les paquets entrant sur l'ordinateur passent dans cette chaîne avant d'avoir passé le routage. • Chaîne INPUT : Tous les paquets destinés à l'ordinateur passent dans cette chaîne. • Chaîne OUTPUT : Tous les paquets créés par l'ordinateur passent dans cette chaîne. • Chaîne FORWARD : Tous les paquets traversant l'ordinateur passent dans cette chaîne. • Chaîne POSTROUTING : Tous les paquets quittant l'ordinateur passent dans cette chaîne après avoir passé le routage.

Les chaînes associées aux différents points d'entrée

Chaîne	Table	Description
PREROUTING	nat, mangle	Par cette chaîne passeront les paquets entrant dans la machine avant routage.
INPUT	filter	Cette chaîne traitera les paquets entrants avant qu'ils ne soient passés aux couches supérieures (les applications).
FORWARD	filter	Ce sont les paquets uniquement transmis par la machine sans que les applications n'en aient connaissance.
OUTPUT	filter, nat, mangle	Cette chaîne sera appelée pour des paquets envoyés par des programmes présents sur la machine.
POSTROUTING	nat	Les paquets prêts à être envoyés (soit transmis, soit générés) seront pris en charge par cette chaîne.

Les cibles prédéfinies les plus courantes :

Cible	Description
ACCEPT	Les paquets envoyés vers cette cible seront tout simplement acceptés et pourront poursuivre leur cheminement au travers des couches réseaux.
DROP	Cette cible permet de jeter des paquets qui seront donc ignorés.
REJECT	Permet d'envoyer une réponse à l'émetteur pour lui signaler que son paquet a été refusé.
LOG	Demande au noyau d'enregistrer des informations sur le paquet courant. Cela se fera généralement dans le fichier /var/log/messages (selon la configuration du programme syslogd).
MASQUERADE	Cible valable uniquement dans la chaîne POSTROUTING de la table nat. Elle change l'adresse IP de l'émetteur par celle courante de la machine pour l'interface spécifiée. Cela permet de masquer des machines et de faire par exemple du partage de connexion.
SNAT	Egalement valable pour la chaîne POSTROUTING de la table nat seulement. Elle modifie aussi la valeur de l'adresse IP de l'émetteur en la remplaçant par la valeur fixe spécifiée.
DNAT	Valable uniquement pour les chaînes PREROUTING et OUTPUT de la table nat. Elle modifie la valeur de l'adresse IP du destinataire en la remplaçant par la valeur fixe spécifiée.
RETURN	Utile dans les chaînes utilisateurs. Cette cible permet de revenir à la chaîne appelante. Si RETURN est utilisé dans une des chaînes de base précédente, cela est équivalent à l'utilisation de sa cible par défaut.

IpTables :

iptables est l'outil qui est fourni à l'administrateur pour agir sur tous les concepts vus précédemment et pour modifier les règles de filtrage donc.

La première option à connaître est -t qui permet de spécifier le nom de la table sur laquelle porteront les autres paramètres. Si cette option n'est pas spécifiée, ce sera par défaut la table filter.

On peut aussi demander à iptables de charger un module particulier avec l'option -m. Ce module peut ajouter de nouvelles tables ou de nouvelles manières de tester les paquets.*

Netfilter :

Netfilter travaille sur des paquets réseaux. Il s'agit de parties des informations transmises. Pour, par exemple, télécharger un fichier, celui-ci est découpé en plusieurs paquets avant de transiter sur le réseau. Chacun de ces paquets comporte en plus des données, des informations ajoutées par les couches réseaux. Ce sont sur ces informations que s'effectueront les tests de filtrage.

La couche réseau Linux présente plusieurs points d'accès. Netfilter dispose de fonctions de rappel (callback). Celles-ci sont des suites d'instructions qui précisent ce qui doit être fait lorsque survient un événement.

Concrètement, lorsqu'un paquet réseau atteint un de ces points d'accès, il est passé à Netfilter par l'intermédiaire de sa fonction de rappel. Il est alors examiné pour prendre une décision concernant son traitement futur.

Netfilter se comporte comme un automate qui compare le paquet successivement à plusieurs règles. Et selon le résultat du test, le paquet est traité ou transmis au test suivant.

II- Empêcher le ping sur l'adresse de loopback

- Ping effectué avant la mise en place de la règle iptables

Du serveurs vers 127.0.0.1 :

```
root@debian:~# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.033 ms
```

Du client vers 127.0.0.1 :

```
C:\WINDOWS\system32>ping 127.0.0.1

Envoi d'une requête 'Ping' 127.0.0.1 avec 32 octets de données :
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128
```

- Etape 1: création d'une chaine personnelle

Iptables -N machaine

```
root@debian:~# iptables -N NO_PING_LOOPBACK
```

- Etape 2: prise en compte de cette chaîne dans les logs

Iptables -A machaine -j LOG

```
root@debian:~# iptables -A NO_PING_LOOPBACK -j LOG
```

- Etape 3: prise en compte de l'action de la chaine

Iptables -A machaine -j DROP

```
root@debian:~# iptables -A NO_PING_LOOPBACK -j DROP
```

- Etape 4: écriture de la chaine

Iptables -A INPUT -p icmp -s 127.0.0.1 -j machaine

```
root@debian:~# iptables -A INPUT -p icmp -s 127.0.0.1 -j NO_PING_LOOPBACK
```

- Etape 5 : vérification de ce que nous avons fait

Le ping sur l'adresse de Loopback est impossible :

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
^C
--- 127.0.0.1 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4084ms
```

- Lister toutes les règles du pare-feu

Iptables -L

+

```

root@debian:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
NO_PING_LOOPBACK  icmp -- localhost            anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain NO_PING_LOOPBACK (1 references)
target     prot opt source                destination
LOG        all  -- anywhere             anywhere             LOG level warning
DROP       all  -- anywhere             anywhere

```

- On affiche le contenu des logs

```
tail -f /var/log/syslog
```

D'un coté on ping l'adresse 127.0.0.1 et dans u,n autre terminal on voit les logs apparaissent

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
^C  
--- 127.0.0.1 ping statistics ---  
5 packets transmitted, 0 received, 100% packet loss, time 4084ms
```

```
Mar 17 18:43:56 debian kernel: [ 6080.763210] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=39227 DF PROTO=ICMP TYPE=8 CODE=0 ID=992 SEQ=1
Mar 17 18:43:57 debian kernel: [ 6081.774750] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=39322 DF PROTO=ICMP TYPE=8 CODE=0 ID=992 SEQ=2
Mar 17 18:43:58 debian kernel: [ 6082.799445] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=39526 DF PROTO=ICMP TYPE=8 CODE=0 ID=992 SEQ=3
Mar 17 18:43:59 debian kernel: [ 6083.823422] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=39602 DF PROTO=ICMP TYPE=8 CODE=0 ID=992 SEQ=4
Mar 17 18:44:00 debian kernel: [ 6084.847456] IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=39670 DF PROTO=ICMP TYPE=8 CODE=0 ID=992 SEQ=5
```

- Pour supprimer une règle

On regarde la chaine avec :

```
iptables -L
```

```
root@debian:~# iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
NO_PING_LOOPBACK icmp -- localhost anywhere
```

On affiche les numéros des lignes des règles iptables de cette chaine :

```
iptables -L chaine --line-numbers
```

```
root@debian:~# iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target      prot opt source                destination
1    NO_PING_LOOPBACK  icmp -- localhost            anywhere
```

Pour supprimer la règle de la ligne 1 :

Iptables -D INPUT 1

```
root@debian:~# iptables -D INPUT 1
```

III- Travail à faire

1. empêcher le ping du poste serveur sur le poste client

Poste serveur :

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
000
    link/ether 08:00:27:cb:e2:84 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.41/24 brd 192.168.1.255 scope global enp0s3
```

Poste client :

```
Carte Ethernet Ethernet :

Suffixe DNS propre à la connexion. . . : lan
Adresse IPv6 de liaison locale. . . . : fe80::8498:81f1:850c:8676%3
Adresse IPv4. . . . . : 192.168.1.19
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.1.254
```

Ping du client vers serveur avant la règle iptables :

```
C:\WINDOWS\system32>ping 192.168.1.41

Envoi d'une requête 'Ping' 192.168.1.41 avec 32 octets de données :
Réponse de 192.168.1.41 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.41 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.41 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.41 : octets=32 temps<1ms TTL=64
```

Ping du serveur vers client avant la règle iptables :

```
root@debian:~# ping 192.168.1.19
PING 192.168.1.19 (192.168.1.19) 56(84) bytes of data.
64 bytes from 192.168.1.19: icmp_seq=1 ttl=128 time=4.65 ms
64 bytes from 192.168.1.19: icmp_seq=2 ttl=128 time=0.311 ms
64 bytes from 192.168.1.19: icmp_seq=3 ttl=128 time=0.343 ms
```

Pour empêcher le client à ping le serveur :

Iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT

```
root@debian:~# iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT
```

Test de ping du client vers le serveur :

```
C:\Users\Windows>ping 192.168.1.41

Envoi d'une requête 'Ping' 192.168.1.41 avec 32 octets de données :
Réponse de 192.168.1.41 : Impossible de joindre le port de destination.
Réponse de 192.168.1.41 : Impossible de joindre le port de destination.
Réponse de 192.168.1.41 : Impossible de joindre le port de destination.
Réponse de 192.168.1.41 : Impossible de joindre le port de destination.

Statistiques Ping pour 192.168.1.41:
    Paquets : envoyés = 4, reçus = 0, perdus = 4 (perte 100%),
```

2. Permettre d'accéder à votre serveur web en http uniquement

Pour autoriser uniquement l'accès au serveur Web en http uniquement il faut bloquer le TCP et l'UDP ainsi que le port 443 (https) en entrée et en sortie.

```
Iptables -A INPUT -p tcp --destination-port 443 -j DROP
```

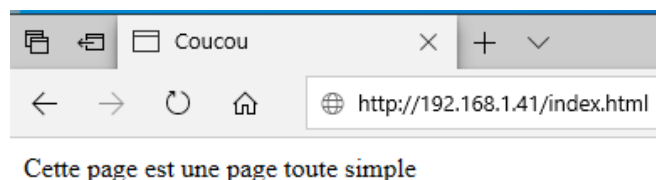
```
Iptables -A OUTPUT -p tcp --destination-port 443 -j DROP
```

```
Iptables -A INPUT -p udp --destination-port 443 -j DROP
```

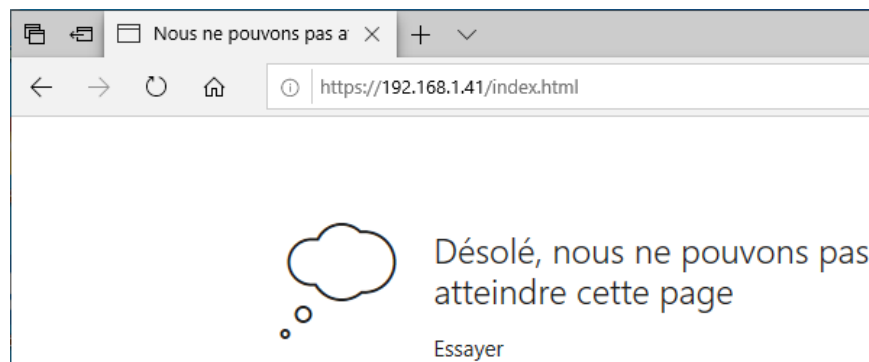
```
Iptables -A OUTPUT -p udp --destination-port 443 -j DROP
```

```
root@debian:~# iptables -A INPUT -p tcp --destination-port 443 -j DROP
root@debian:~# iptables -A OUTPUT -p tcp --destination-port 443 -j DROP
root@debian:~# iptables -A INPUT -p udp --destination-port 443 -j DROP
root@debian:~# iptables -A OUTPUT -p udp --destination-port 443 -j DROP
```

Test d'accès au serveur Web en http : possible



Test d'accès au serveur Web en https : impossible



3. Interdire l'accès à une seule adresse

Client Windows (2 cartes réseaux) : IP 1 : 192.168.1.19 /24 / IP 2 : 192.168.1.12 /24

Serveur : 192.168.1.41 /24

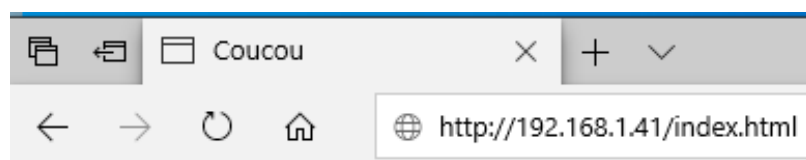
Il faut bloquer en entrée du serveur la deuxième adresse IP du client pour interdire l'accès de celle-ci sur le serveur.

Iptables -A INPUT -s 192.168.1.12 -j DROP

```
root@debian:~# iptables -A INPUT -s 192.168.1.12 -j DROP
```

Test avec la règle iptables : avec l'IP 1 on a accès au serveur Web en http.

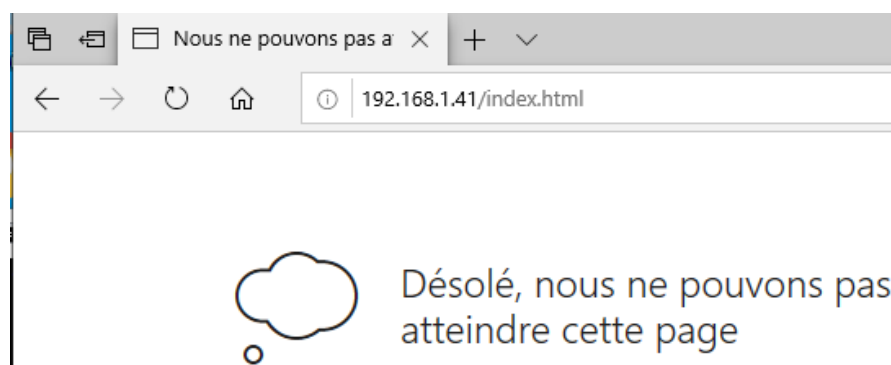
```
Carte Ethernet Ethernet :  
  
Suffixe DNS propre à la connexion. . . : lan  
Adresse IPv6 de liaison locale. . . . : fe80::8498:81f1:850c:8676%3  
Adresse IPv4. . . . . : 192.168.1.19  
Masque de sous-réseau. . . . . : 255.255.255.0  
Passerelle par défaut. . . . . : 192.168.1.254
```



Cette page est une page toute simple

Test après la règle iptables : avec l'IP 2 on n'a pas accès au serveur Web en http.

```
Carte Ethernet Ethernet 2 :  
  
Suffixe DNS propre à la connexion. . . : lan  
Adresse IPv6 de liaison locale. . . . : fe80::71e3:4e59:ae29:8762%7  
Adresse IPv4. . . . . : 192.168.1.12  
Masque de sous-réseau. . . . . : 255.255.255.0  
Passerelle par défaut. . . . . : 192.168.1.254
```



4. Refuser toute connexion telnet (port 23)

Il faut refuser les connexions entrantes sur le serveur avec le port 23 (telnet).

Iptables -A INPUT -p tcp --destination-port 23 -j DROP

```
root@debian:~# iptables -A INPUT -p tcp --destination-port 23 -j DROP ..
```

Test de connexion telnet depuis le client : Impossible

```
C:\Users\Windows>telnet 192.168.1.41
Connexion à 192.168.1.41...Impossible d'ouvrir une connexion à l'hôte, sur le port 23: Échec lors de la connexion
```

5. Ecrire les règles qui répondent aux demandes suivantes

- Votre poste client ne peut que consulter votre serveur web
On autorise en entrée et en sortie le poste client à accéder au serveur Web sur le port 80 (http)

Iptables -A INPUT -s 192.168.1.19 -p tcp --dport 80 -j ACCEPT

Iptables -A OUTPUT -s 192.168.1.19 -p tcp --dport 80 -j ACCEPT

```
root@debian:~# iptables -A INPUT -s 192.168.1.19 -p tcp --dport 80 -j ACCEPT
root@debian:~# iptables -A OUTPUT -s 192.168.1.19 -p tcp --dport 80 -j ACCEPT
```

- Le poste client ne peut pas pinguer votre serveur
On bloque l'adresse IP du client en entrée du serveur Web avec le protocole icmp

Iptables -A INPUT -s 192.168.1.19 -p icmp --icmp-type echo-request -j REJECT

```
root@debian:~# iptables -A INPUT -s 192.168.1.19 -p icmp --icmp-type echo-request -j REJECT
```

Test de ping du client vers le serveur : impossible

```
C:\Users\Windows>ping 192.168.1.41

Envoi d'une requête 'Ping' 192.168.1.41 avec 32 octets de données :
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
```

- Le poste client ne peut pas être pingué
On désactive le ping en sortie sur l'adresse IP du client avec le protocole icmp (ping)

Iptables -A OUTPUT -p icmp -d 192.168.1.19 -j DROP

```
root@debian:~# iptables -A OUTPUT -p icmp -d 192.168.1.19 -j DROP
```

Test de ping du serveur vers le client : impossible

```
root@debian:~# ping 192.168.1.19
PING 192.168.1.19 (192.168.1.19) 56(84) bytes of data.
ping: sendmsg: Opération non permise
ping: sendmsg: Opération non permise
ping: sendmsg: Opération non permise
```

- Votre serveur web est uniquement serveur web

On accepte les connexion entrante et sortante avec le port 80 (http) pour le serveur Web.

Iptables -A INPUT -p tcp --dport 80 -j ACCEPT

Iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT

```
root@debian:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@debian:~# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
```

Test de ping du client vers le serveur : impossible

```
C:\Users\Windows>ping 192.168.1.41

Envoi d'une requête 'Ping' 192.168.1.41 avec 32 octets de données :
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.

Statistiques Ping pour 192.168.1.41:
    Paquets : envoyés = 2, reçus = 0, perdus = 2 (perte 100%),
```

- Seules les connexions établies sont acceptées

Avec la première ligne les connexions seront fermées et on n'aura plus accès à notre machine. On doit donc ajouter une autre commande qui permet de ne pas fermer les connexions déjà établies.

Iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

Iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

```
root@debian:~# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
root@debian:~# iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Pour lister toutes les règles iptables : iptables -L

```
root@debian:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            tcp dpt:https
DROP       tcp  --  anywhere              anywhere               udp dpt:443
DROP       udp  --  anywhere              anywhere
DROP       all  --  DESKTOP-K1IEEGA-001   anywhere
DROP       tcp  --  anywhere              anywhere               tcp dpt:telnet
ACCEPT     tcp  --  DESKTOP-K1IEEGA       anywhere               tcp dpt:http
REJECT     icmp --  DESKTOP-K1IEEGA       anywhere               icmp echo-request reject-with icmp-port-unreachable
REJECT     icmp --  anywhere              anywhere               icmp echo-request reject-with icmp-port-unreachable
DROP       icmp --  anywhere              DESKTOP-K1IEEGA
ACCEPT     tcp  --  anywhere              anywhere               tcp dpt:http
ACCEPT     all  --  anywhere              anywhere               state RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination            tcp dpt:https
DROP       tcp  --  anywhere              anywhere               udp dpt:443
ACCEPT     tcp  --  DESKTOP-K1IEEGA       anywhere               tcp dpt:http
REJECT     icmp --  DESKTOP-K1IEEGA       anywhere               icmp echo-request reject-with icmp-port-unreachable
DROP       icmp --  anywhere              DESKTOP-K1IEEGA
ACCEPT     tcp  --  anywhere              anywhere               tcp dpt:http
ACCEPT     all  --  anywhere              anywhere               state RELATED,ESTABLISHED
```

Adresse IP client :

Carte Ethernet Ethernet :

```
Suffixe DNS propre à la connexion. . . : lan
Adresse IPv6 de liaison locale. . . . : fe80::8498:81f1:850c:8676%3
Adresse IPv4. . . . . : 192.168.1.19
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.1.254
```

Carte Ethernet Ethernet 2 :

```
Suffixe DNS propre à la connexion. . . : lan
Adresse IPv6 de liaison locale. . . . : fe80::71e3:4e59:ae29:8762%7
Adresse IPv4. . . . . : 192.168.1.12
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.1.254
```

Adresse IP serveur :

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:cb:e2:84 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.41/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feeb:e284/64 scope link
        valid_lft forever preferred_lft forever
```