TP Filtrage réseau avec Iptables LUCA BONA

Les paramètres IPTables

Les actions sur les chaînes (Commandes)

- -A : on ajoute (Append) une règle à la fin d'une chaîne existante.
- -P : on définit la politique par défaut d'une chaîne (Policy). C'est ce qui s'applique si le paquet ne correspond à aucune règle.
- -N : on crée une nouvelle chaîne totalement personnalisée (New).
- -F : on efface (Flush) toutes les règles d'une chaîne pour remettre à zéro.
- -L : on liste les règles actuellement actives pour vérifier notre configuration.

Les cibles de flux (Chaînes)

- INPUT : on cible les paquets qui rentrent dans la machine (la destination finale du paquet, c'est cette machine).
- OUTPUT : on cible les paquets qui sortent de la machine (c'est cette machine qui crée et envoie le paquet).
- FORWARD : on cible les paquets qui ne font que traverser la machine, typiquement le cas d'un routeur.
- POSTROUTING : on cible les paquets juste avant qu'ils ne sortent (utilisé spécifiquement pour le NAT/Masquerade).

Les filtres de base (Règles)

- -p : on cible un protocole particulier, comme tcp, udp ou icmp (pour le ping).
- --dport : on cible le port de destination, par exemple 80 pour le trafic web ou 22 pour le trafic SSH.
- -s : on cible l'adresse IP source, c'est-à-dire l'expéditeur du paquet.
- -d : on cible l'adresse IP de destination, c'est-à-dire le destinataire du paquet.
- -i : on cible l'interface réseau par laquelle le paquet entre (in), par exemple ens20.
- -o : on cible l'interface réseau par laquelle le paquet sort (out).

Les décisions (Cibles / Jump)

- -j ACCEPT : on autorise le paquet à passer et on arrête de lire les règles suivantes.
- -j DROP : on détruit le paquet silencieusement, il est ignoré et l'expéditeur ne reçoit pas de message d'erreur.

- -j LOG : on enregistre les informations du paquet dans le journal système (syslog), puis le pare-feu passe à la règle d'en dessous.
- -j MASQUERADE : on remplace l'adresse IP source privée par l'adresse IP publique de la machine (utilisé pour donner internet au LAN).

Les modules avancés (Extensions)

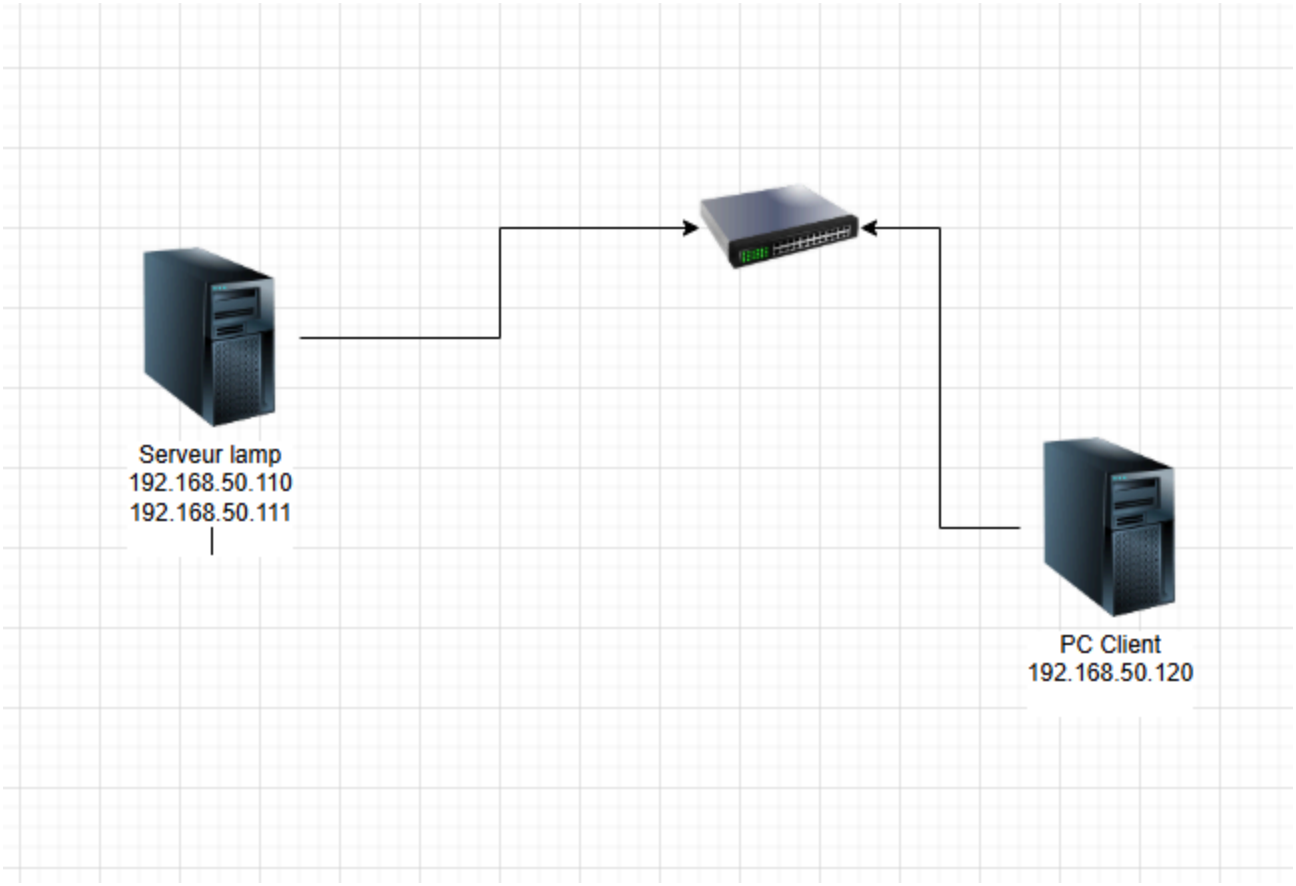
- -t : on sélectionne la table à utiliser, par exemple "nat" pour modifier les adresses IP. Si on ne met rien, c'est la table "filter" par défaut.
- -m : on charge une extension spécifique pour avoir accès à des options de filtrage plus fines.
- --state ESTABLISHED,RELATED : on cible l'état du paquet. Ici, on autorise les paquets qui font partie d'une connexion déjà ouverte (le trafic de retour).
- --mac-source : utilisé avec le module mac (-m mac), on cible l'adresse physique exacte d'une carte réseau, indépendamment de son adresse IP.

Découverte IPTables

1. Architecture du réseau

Le réseau mis en place pour cette phase de découverte est très simple et ne comprend que deux machines.

- Un Poste 1 : Il fait office de serveur web (type Apache) et de serveur SSH.
- Un Poste 2 : Il s'agit de la station de travail, qui fera office de client.



2. Rappel sur la structure d'une commande

La majorité des commandes IPTables respectent la syntaxe suivante :

`iptables [-t <nom-table>] <commande> <nom-chaîne> <paramètre-1> <option-1>`.

Les éléments clés à retenir sont :

- TABLE : Il existe 3 tables possibles pour Netfilter, la table par défaut étant la table Filter.
- CHAINE : Permet d'affiner le filtrage. On utilise INPUT pour les paquets entrants, OUTPUT pour les paquets sortants, et FORWARD pour la passerelle.
- CIBLE : Indique l'action à réaliser sur le paquet (accepter, refuser, rejeter, etc.).
- POLICE : Définit le comportement par défaut pour les paquets qui ne correspondent à aucune règle spécifique.

3. Exercice : Empêcher le ping sur la boucle locale (Loopback)

L'objectif est de bloquer et de tracer les requêtes de ping envoyées à la machine elle-même (127.0.0.1).

1. Création d'une chaîne personnelle :

Bash

```
iptables -N Lchaine
```

Cette commande crée une nouvelle chaîne de filtrage nommée "machaine".

2. Ajout de la journalisation et de l'action de blocage :

Bash

```
iptables -A Lchaine -j LOG
iptables -A Lchaine -j DROP
```

Ces règles indiquent que tout paquet entrant dans "machaine" sera d'abord écrit dans les logs , puis détruit.

```
root@debian12:~# iptables -N Lchaine
root@debian12:~# iptables -A Lchaine -j LOG
root@debian12:~# iptables -A Lchaine -j DROP
```

3. Application de la chaîne au trafic ICMP local :

Bash

```
iptables -A INPUT -p icmp -s 127.0.0.1 -j Lchaine
```

Cette règle capture le ping (icmp) provenant de 127.0.0.1 et l'envoie vers notre chaîne personnalisée.

```
root@debian12:~# iptables -A INPUT -p icmp -s 127.0.0.1 -j Lchaine
```

Pour tester, on effectue un ping sur l'adresse de loopback dans un terminal, et on observe le résultat de la commande `tail -f /var/log/syslog` dans un second terminal. Les logs doivent afficher des informations sur les paquets bloqués.

```
root@debian12:~# ping 127.0.0.0
PING 127.0.0.0 (127.0.0.0) 56(84) bytes of data.
```

```
mars 11 18:21:21 debian kernel: PING_BLOCKED: IN=lo OUT= MAC=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=33411 DF PROTO=ICMP TYPE=8 CODE=0 ID=44065 SEQ=57
```

- `mars 11 18:21:21` : Date et heure exactes de l'événement.

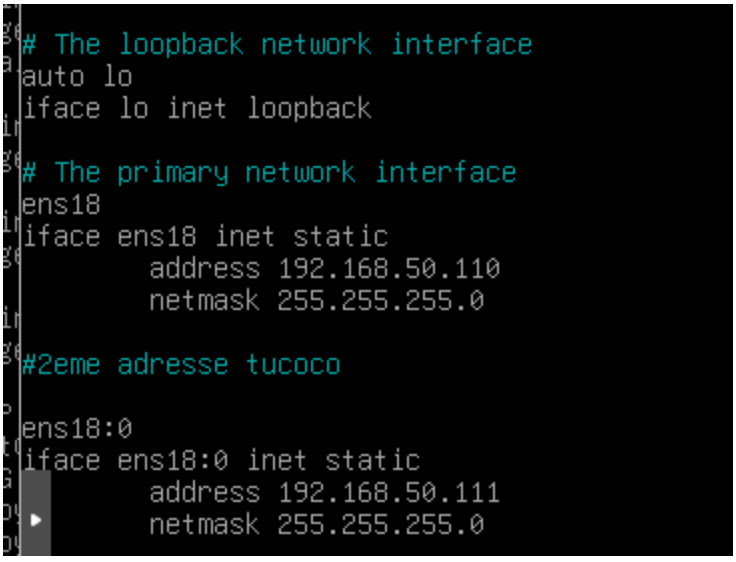
- `debian kernel:` : Indique que le message a été généré directement par le noyau Linux (qui gère le pare-feu Netfilter).
- `PING_BLOCKED:` : C'est le préfixe personnalisé (défini par l'option `--log-prefix`) qui permet à l'administrateur d'identifier rapidement pourquoi la ligne a été générée.

- `IN=lo OUT=` : Interfaces réseau concernées. Ici, le paquet entre par la boucle locale (lo). Le champ OUT est vide car le paquet n'est pas redirigé vers l'extérieur.
- `MAC=00:00:00:00:00:00:00:00:00:00:08:00` : Adresse matérielle de la carte réseau. Elle est remplie de zéros ici car le trafic reste interne (interface lo).
- `SRC=127.0.0.1 DST=127.0.0.1` : Adresses IP source et destination. Cela confirme que la machine s'envoie un paquet à elle-même.
- `LEN=84` : Taille totale du paquet réseau, exprimée en octets.
- `TOS=0x00 PREC=0x00` : Type of Service et Précédence. Ces champs gèrent la priorité de traitement du paquet (Qualité de Service).
- `TTL=64` : Time-To-Live (Durée de vie). C'est le nombre maximum de routeurs que le paquet est autorisé à traverser avant d'être détruit pour éviter qu'il ne tourne en boucle.
- `ID=33411 DF` : Identifiant unique du paquet IP et drapeau "Don't Fragment" (indique que le routeur ne doit pas découper ce paquet s'il est trop gros pour le réseau suivant).
- `PROTO=ICMP` : Protocole utilisé. L'ICMP (Internet Control Message Protocol) est le protocole de diagnostic réseau utilisé par le ping.
- `TYPE=8 CODE=0` : Détail de l'entête ICMP. Le type 8 correspond précisément à une requête de ping (Echo Request). Le code 0 indique qu'il s'agit d'une requête standard sans sous-catégorie.
- `ID=44065 SEQ=57` : Identifiant de session et numéro de séquence du processus ping, permettant d'associer précisément la requête à sa réponse.

4. Configuration complète

L'objectif final est de configurer les deux postes de manière très stricte selon un cahier des charges précis.

Configuration du post client



Configuration du Poste Serveur (Web)

Fichier /etc/network/interfaces du serveur:

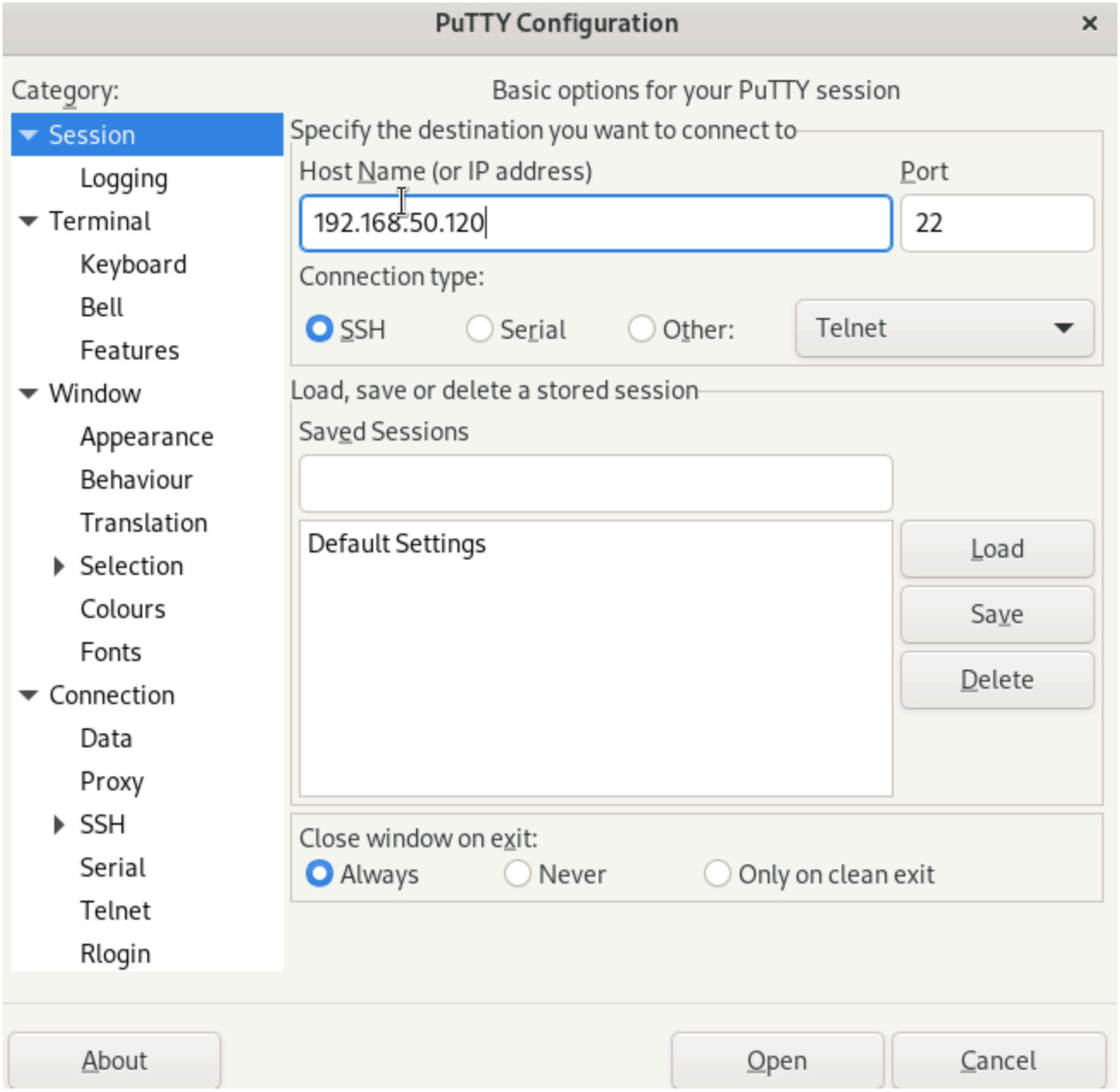
Plaintext

```
auto enp0s18
iface enp0s18 inet static
    address 192.168.50.120
    netmask 255.255.255.0
```

Règles IPTables sur le serveur: Le serveur doit être uniquement un serveur web , il ne doit pas pouvoir pinguer le client , il doit interdire l'accès à la seconde adresse IP du client , et n'accepter que les connexions établies.

Bash

```
# 1. Politiques par défaut
iptables -P INPUT DROP
iptables -P OUTPUT DROP

# 2. Autoriser uniquement le retour des connexions établies
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# 3. Bloquer l'accès depuis la seconde adresse IP du client
iptables -A INPUT -s 192.168.50.111 -j DROP

# 4. Autoriser uniquement le trafic Web (HTTP)
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# 5. Empêcher le serveur de pinguer le client (par sécurité supplémentaire,
bien que bloqué par défaut ici)
iptables -A OUTPUT -d 192.168.50.120 -p icmp -j DROP

# 6. on bloque le port 22
iptables -A INPUT -p tcp -dport 22 -j DROP

# 7. on bloque le port 23 telnet
iptables -A INPUT -p tcp -dport 23 -j DROP
```



Le ssh ne marche pas

“Capture d'écran 2026-03-14 200409.png” ne peut être trouvé.

Pareil pour le telnet :

```
root@debian12:~# telnet 192.168.50.120
Trying 192.168.50.120...
telnet: Unable to connect to remote host: Connexion terminée par expiration du délai d'attente
```

Test ping :

```
root@debian12:~# iptables -A OUTPUT -d 192.168.50.120 -p icmp --icmp-type echo-request -j DROP
root@debian12:~# ping 192.168.50.120
PING 192.168.50.120 (192.168.50.120) 56(84) bytes of data.
```

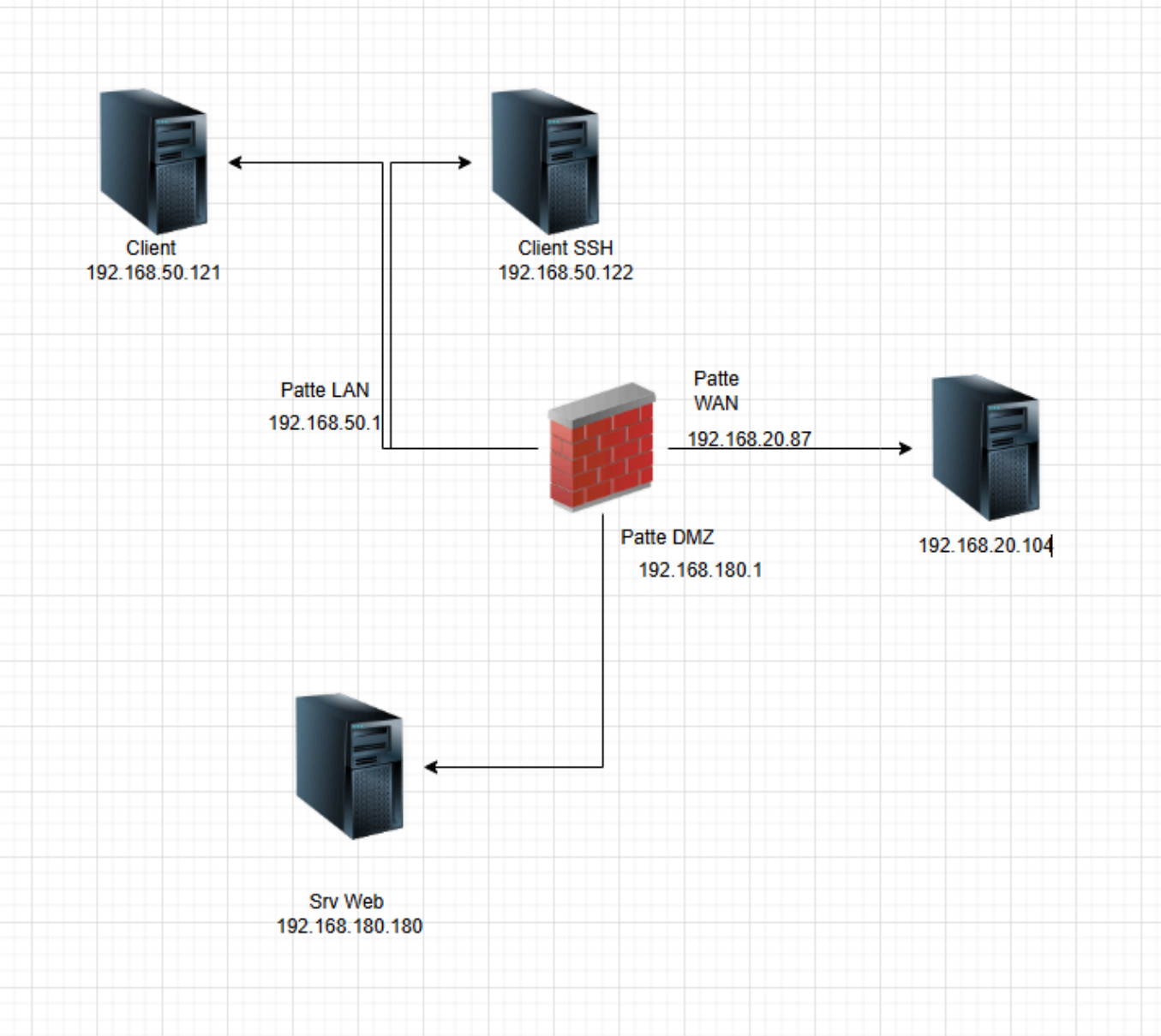
Test web :

TP SUIVANT

1.Plan d'adressage IP :

Machine	IP
1. Router	Patte WAN : 192.168.20.87 (salle de cours) Patte LAN : 192.168.50.1 Patte DMZ : 192.168.180.1
2. PC Client	192.168.50.121
3. Client SSH	192.168.50.122
4. SRVWEB DMZ	192.168.180.180
5. Client WAN	192.168.20.104

SCHEMA RESEAU :



2. Architecture du réseau

Pour sécuriser l'infrastructure, on la divise en trois zones distinctes :

Zone	Interface (Exemple)	Rôle
WAN	ens20	Internet. La zone publique non sécurisée.
DMZ	enp0s18	Zone tampon. Héberge les serveurs accessibles de l'extérieur (Web, DNS).
LAN	enp0s19	Réseau privé. Contient les postes de travail et le serveur SSH.

3. Prérequis : Activer le routage

Par défaut, Linux ne laisse pas passer les paquets d'une carte réseau à une autre. Il faut activer l'IP forwarding pour transformer la machine en routeur.

Bash

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

> Le but : Le noyau Linux accepte désormais de faire transiter les données entre le LAN, la DMZ et le WAN.

4. Configuration des règles IPTables

Les commandes suivantes sont à taper sur le routeur. L'ordre est très important.

A. Initialisation et maintien d'état

On commence par tout bloquer par défaut, puis on gère le retour des paquets.

Bash

```
iptables -P FORWARD DROP
```

- Rôle : Politique par défaut. Si un paquet traversant le routeur ne correspond à aucune règle, il est détruit.

Bash

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

- Rôle : On n'accepte que les connexions établies. Si une machine du réseau interne initie une requête, cette règle autorise automatiquement la réponse à rentrer. Sans ça, le réseau est bloqué au retour.

B. Règles pour la DMZ

On gère l'accès aux serveurs exposés.

Bash

```
iptables -A FORWARD -d 192.168.180.180 -p tcp --dport 80 -j ACCEPT
```

- Rôle : Rend le serveur Web de la DMZ accessible de partout (Internet et LAN) sur le port HTTP.

Bash

```
iptables -A FORWARD -s 192.168.50.0/24 -d 192.168.180.180 -p icmp -j ACCEPT
```

- Rôle : Autorise le ping depuis le LAN vers la DMZ pour vérifier la connectivité. (À placer avant la règle de blocage globale du ping).

Bash

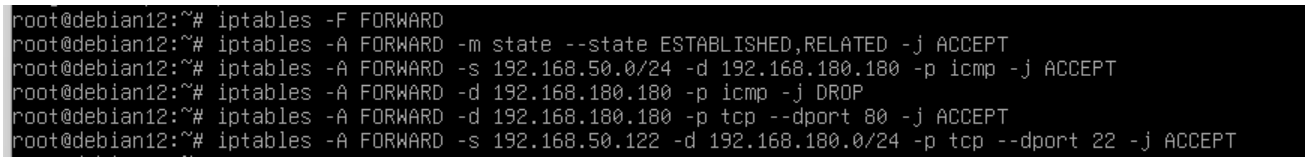
```
iptables -A FORWARD -i ens20 -d 192.168.180.180 -p icmp -j DROP
```

- Rôle : Empêche les attaques DDoS en interdisant le ping venant de l'extérieur (Internet/WAN) vers la DMZ.

Bash

```
iptables -A FORWARD -s 192.168.50.122 -d 192.168.180.0/24 -p tcp --dport 22 -j ACCEPT
```

- Rôle : Autorise le serveur SSH du LAN (ici 192.168.50.92) à se connecter aux machines de la DMZ pour l'administration.



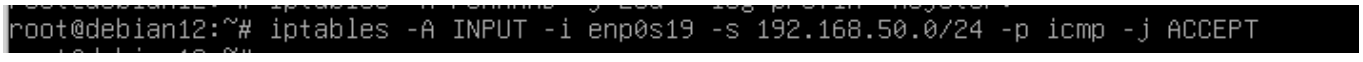
C. Règles pour le LAN

On gère les droits des utilisateurs du réseau privé.

Bash

```
iptables -A INPUT -i enp0s19 -s 192.168.50.0/24 -p icmp -j ACCEPT
```

- Rôle : Autorise le ping de tous les postes du LAN vers l'interface LAN du routeur. On utilise INPUT car le paquet s'arrête au routeur.



Bash

```
iptables -A FORWARD -s 192.168.50.0/24 -d 192.168.180.0/24 -j ACCEPT
```

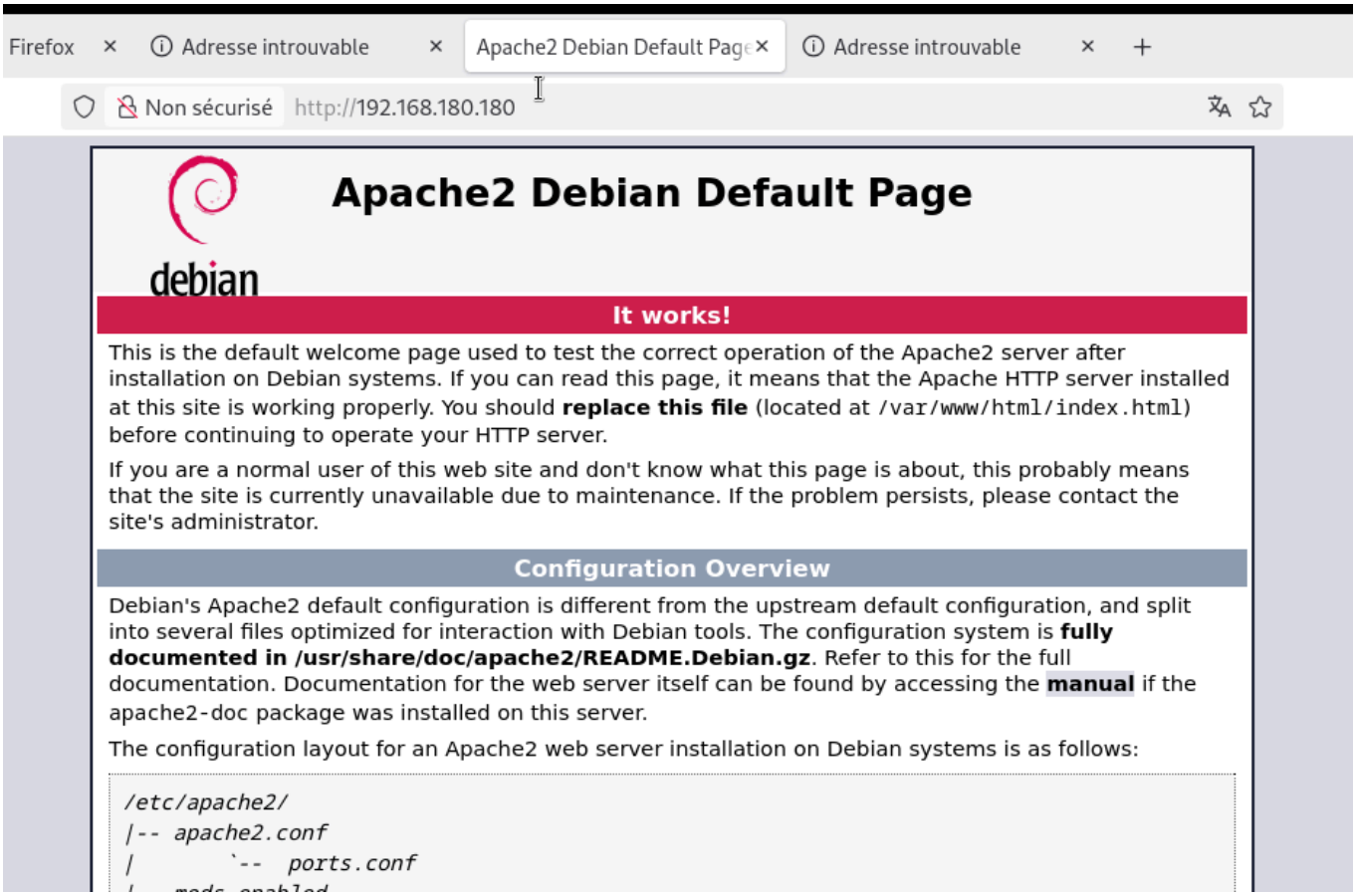
- Rôle : Autorise le routage classique des paquets du LAN vers la DMZ.

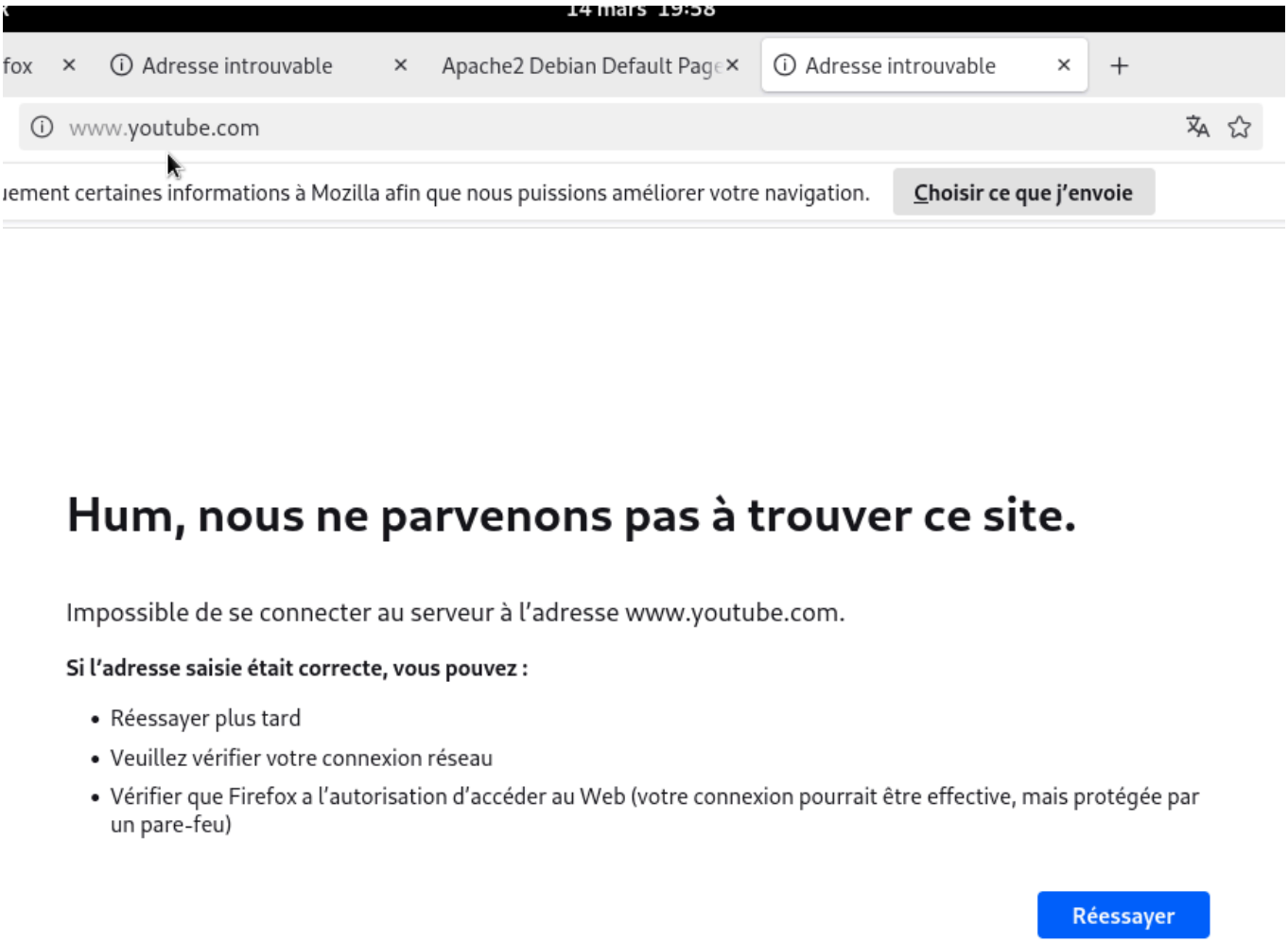
```
root@debian12:~# iptables -A FORWARD -s 192.168.50.0/24 -d 192.168.180.0/24 -j ACCEPT
```

Bash

```
iptables -A FORWARD -m mac --mac-source BC:24:11:28:F4:2C -o enp0s20 -j DROP
```

- Rôle : Empêche le poste du comptable d'aller sur Internet en filtrant directement son adresse physique (MAC).





Bash

```
iptables -t nat -A POSTROUTING -s 192.168.50.0/24 -o ens20 -j MASQUERADE
```

- Rôle : Camoufle les adresses IP privées du LAN derrière l'adresse IP publique du routeur pour permettre l'accès à Internet (NAT).

“Capture d'écran 2026-03-14 181008.png” ne peut être trouvé.

D. Journalisation

Bash

```
iptables -A FORWARD -j LOG --log-prefix "REJETER_TP: "
```

- Rôle : Tout paquet qui n'a pas validé les autorisations précédentes arrive ici. Il est enregistré dans les logs du système avant d'être refusé par la politique globale DROP.

4. Tableau de routage final

   Afficher les règles actives pour vérifier qu'elles sont dans le bon ordre.

Bash

```
iptables -L -n -v
```

```
root@debian12:~# iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     1    --  enp0S19 *       192.168.50.0/24      0.0.0.0/0
    0     0 ACCEPT     1    --  enp0s19 *       192.168.50.0/24      0.0.0.0/0

Chain FORWARD (policy DROP 71481 packets, 5191K bytes)
 pkts bytes target     prot opt in     out     source               destination
  913 97139 ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    2   168 ACCEPT     1    --  *      *       192.168.50.0/24      192.168.180.180
    0     0 DROP       1    --  *      *       0.0.0.0/0            192.168.180.180
    2   120 ACCEPT     6    --  *      *       0.0.0.0/0            192.168.180.180     tcp dpt:80
    0     0 ACCEPT     6    --  *      *       192.168.50.122       192.168.180.0/24    tcp dpt:22
    0     0 ACCEPT     0    --  *      *       192.168.50.0/24      192.168.180.0/24
    0     0 DROP       0    --  *      enp0s20  0.0.0.0/0            0.0.0.0/0           MAC bc:24:11:28:f4:2c
 8124  585K LOG        0    --  *      *       0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 4 prefix "Rejeter:"
    0     0 ACCEPT     0    --  *      *       192.168.50.0/24      192.168.180.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Test ssh :

```
root@debian12-graphique:/home/sio# ssh root@192.168.180.180
The authenticity of host '192.168.180.180 (192.168.180.180)' can't be establishe
d.
ED25519 key fingerprint is SHA256:snDFkU8WUd9kHBooU2pLVp3hoNATCzhDnyRplCiISLE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.180.180' (ED25519) to the list of known host

root@192.168.180.180's password:
Linux debian12 6.1.0-39-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.148-1 (2025-08-2
5) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar 13 22:03:38 2026
root@debian12:~#
```