IP-tables.md

TP-Firewall-IPtables THOMAS GRZESINSKI

Iptables est un outil de filtrage de paquets pour Linux qui permet de configurer le pare-feu du système. Il sert à contrôler le trafic réseau entrant et sortant en définissant des règles basées sur des critères tels que l'adresse IP, le port et le protocole. En résumé, iptables permet de sécuriser un système en autorisant ou en bloquant des connexions réseau selon des règles prédéfinies.

Les avantages et incovénients de IPtables

- Les avantages d'iptables sont bien nombreux, en effet :
  - Nous avons une trés bonne flexibilité ou on peut définor des règles très spécifiques pour le filtrage du trafic
  - Il peut prendre en charge des fonctonnalités avancés comme la gestion des états de connexion (conntrack)
  - Le service est gratuit
  - On peut écrire des scripts pour automatiser la gestion des règles
  - Fonctionne avec le noyau Linux donc il offre des performances optimales
- Mais il y a aussi des incovénients comme :
  - Le logiciel est conplexe et peut-être diffcile à prendre en main au début
  - Si à un moment donné il y a trop de règles on peut vite se perdre
  - Il n'y a pas d'interface graphique qui pourrait facilité la prise en main comme avec pfSense
  - d'instructions qui précisent ce qui doit être fait lorsque survient un événement.

Tables principales de Netfilter

- filter : Permet le filtrage des paquets. En effet elle est utilisé pour autoriser ou bloquer le trafic réseau en fonction des régles spécifiques.
- nat : Permet pour la traduction réseau, permettant de modifier les adresses IP et les ports des paquets afin de gérer des fonctions comme le masquage d'adresses et la redirection des ports.
- mangle : Permet de moficier les paquets en cours de transit, permettant des actions telles que le marquage de paquets, la modificationd es options IP et l'application de politiques de routage spécifiques.

Les chaînes associées aux différents points d'entrée

Un paquet atteignant un de ces points sera envoyé vers la chaîne associée. Chaque chaîne effectue des tests. Ces chaînes sont en nombre fini et ne s'appliquent qu'à certaines tables. Le tableau suivant les liste, en indiquant quelle table a une chaîne de ce type.

Chaine	Table	Description
PREROUTING	filter	Cette chaîne est utilisée pour manipuler les paquets entrant avant le routage. Elle permet d'appliquer des règles sur les paquets avant qu'ils ne soient
INPUT	filter	Cette chaîne traite les paquets qui sont destinés à la machine locale. Elle permet d'appliquer des règles sur les paquets entrants à destination de la n
FORWARD	filter	Cette chaîne est utilisée pour les paquets qui transitent par la machine, c'est-à-dire les paquets qui ne sont ni destinés à la machine elle-même ni gé
OUTPUT	filter	Cette chaîne traite les paquets générés par la machine locale avant qu'ils ne soient envoyés sur le réseau. Elle permet de filtrer et de manipuler les pa
POSTROUTING	nat	Cette chaîne est utilisée pour modifier les paquets après le routage, généralement pour appliquer des règles de translation d'adresses (NAT) sur les p

Les cibles prédéfinies les plus courantes :

Dans le cadre de la gestion du filtrage et du routage des paquets, plusieurs cibles prédéfinies permettent de spécifier le traitement à appliquer aux paquets en fonction des conditions définies dans les chaînes, offrant ainsi une flexibilité dans la configuration des règles de sécurité réseau.

Cible	Description
ACCEPT	Accepte le paquet et permet son passage. Cela signifie que le traitement se termine ici.
DROP	Rejette le paquet sans envoyer de message de retour à l'expéditeur. Le paquet est simplement ignoré.
REJECT	Rejette le paquet et envoie un message d'erreur à l'expéditeur, indiquant que le paquet a été bloqué.
LOG	Enregistre des informations sur le paquet dans les journaux du système, puis le paquet est traité par la cible suivante.
MASQUERADE	Modifie l'adresse IP source d'un paquet sortant pour celle de l'interface sortante. Utilisé généralement pour le partage de connexion.
SNAT	Source Network Address Translation : modifie l'adresse IP source d'un paquet sortant selon des règles définies.
DNAT	Destination Network Address Translation : modifie l'adresse IP de destination d'un paquet entrant selon des règles définies.
RETURN	Permet de revenir à la chaîne appelante après avoir exécuté la chaîne utilisateur. Cela permet de continuer le traitement dans la chaîne initiale.

TP Découverte IPtables

Schéma de notre infrastructure réseau

Mise en place de l'infrastructure réseau

Nous allons tout d'abord installé sur notre machine serveur (Débian 12) un serveur LAMP et un serveur SSH.

Les commandes pour le serveur LAMP :

- apt update et apt install apache2 :
- Apt update pour mettre à jours vos paquets et enuite apt install apache 2. Démarrer ensuite les services apache2 avec la commande systemctl start apache 2 et vérifier le statut de votre web serveur grace la commande systemctl status apache2

```
root@debian:~# systemctl start apache2
root@debian:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-03-11 17:17:21 CET; 1min 2s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1282 (apache2)
      Tasks: 55 (limit: 2305)
     Memory: 11.4M
        CPU: 143ms
     CGroup: /system.slice/apache2.service
             ├─1282 /usr/sbin/apache2 -k start
             ├─1284 /usr/sbin/apache2 -k start
             └─1285 /usr/sbin/apache2 -k start

mars 11 17:17:21 debian systemd[1]: Starting apache2.service - The Apache HTTP Server...
mars 11 17:17:21 debian apachectl[1281]: AH00558: apache2: Could not reliably determine the server's fully qualified dom
mars 11 17:17:21 debian systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

Les commandes pour le serveur SSH:

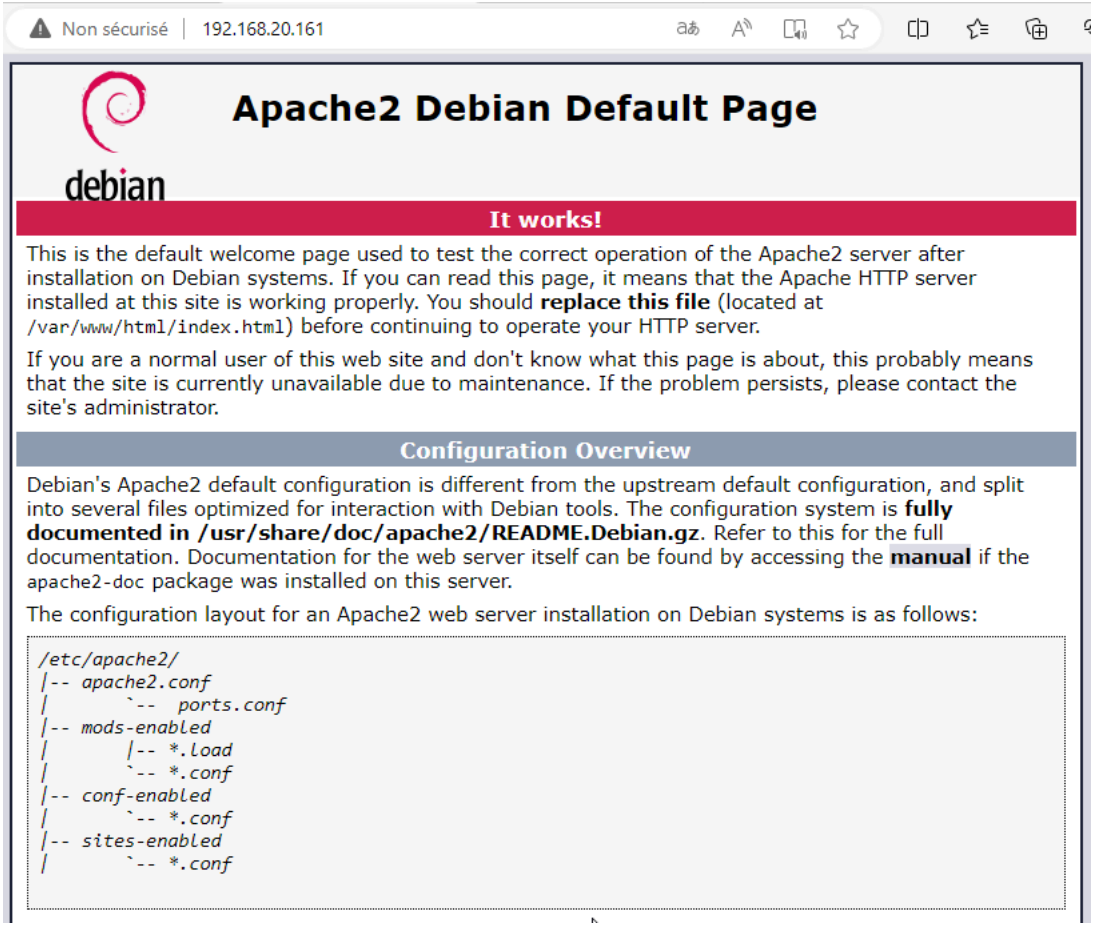
- Apt install SSH alt text
- Pinger vos deux machines pour vérifier leurs communcations
    - Serveur vers client :

```
root@debian:~# ping 192.168.20.115
PING 192.168.20.115 (192.168.20.115) 56(84) bytes of data.
64 bytes from 192.168.20.115: icmp_seq=1 ttl=64 time=2.88 ms
64 bytes from 192.168.20.115: icmp_seq=2 ttl=64 time=1.49 ms
64 bytes from 192.168.20.115: icmp_seq=3 ttl=64 time=1.47 ms
```

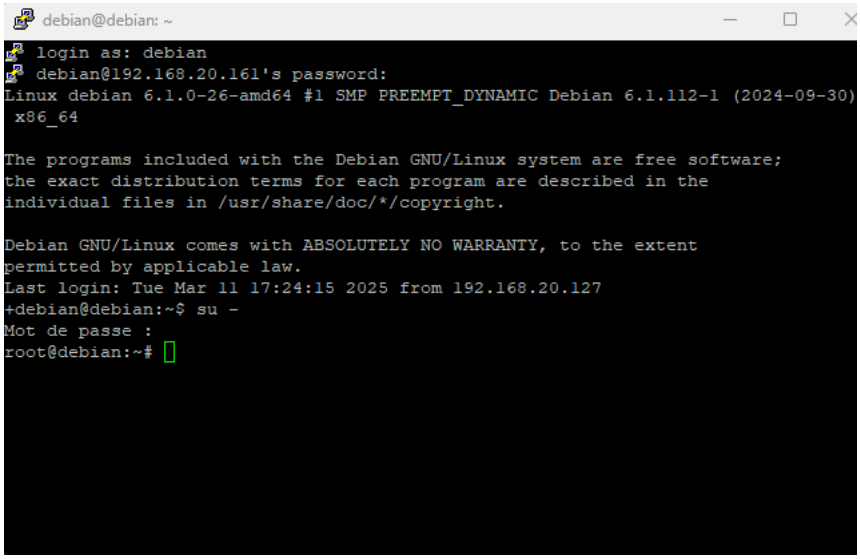
    - Client vers web :

```
C:\Users\sio>ping 192.168.20.161

Envoi d'une requête 'Ping'  192.168.20.161 avec 32 octets de données :
Réponse de 192.168.20.161 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.20.161 : octets=32 temps<1ms TTL=64
Réponse de 192.168.20.161 : octets=32 temps<1ms TTL=64
Réponse de 192.168.20.161 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.20.161:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
```

    vous pouvez aussi accéder au serveur via internet en entrant l'ip de votre serveur



ou bien alors aller dans putty entrer l'ip de votre machine serveur et connectez-vous



Empêcher le ping sur l'adresse de loopback

- Installé iptables avec les commandes apt update et apt-get install iptables
- Nous allons maintenant empecher que le serveur ne puisse se ping lui-même (loopback)
- Configuration:
  - Etape 1 : Nous allons créer une chaine personnel avec la commande iptables -N machaine

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

Explciation: Une chaîne personnalisée permet de regrouper des règles spécifiques pour mieux organiser et gérer vos configurations.

  - Etape 2 : Nous allons maintenant prendre en compte cette chaine dans les logs

```
root@debian:~# iptables -A machaine -j LOG --log-prefix "PING_BLOCKED: " --log-l
evel 4
```

Explication: L'option -A machaine ajoute une règle dans la chaîne spécifiée, tandis que -j LOG enregistre le paquet dans les journaux du système. Le préfixe --log-prefix "PING_BLOCKED: " facilite l'identification des paquets bloqués, et --log-level 4 indique que ces paquets seront notés comme des avertissements. En résumé, cette règle enregistre les paquets ICMP, comme les pings, en les marquant pour une identification facile dans les journaux

- Etape 3 : Nous allons prendre maintenant en compte l'action de la chaine.

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

Explication : Pour bloquer les paquets ICMP (ping), la règle -j DROP rejette silencieusement tout paquet qui atteint cette étape, empêchant ainsi le ping.

- Etape 4 : Ecriture de la chaine

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

Explication : ette commande redirige tous les paquets ICMP (ping) provenant de l'adresse 127.0.0.1 vers la chaîne machaine, où ils seront enregistrés et bloqués.

- Etape 5 : Vérification
- Listez toutes les règles actives pour vérifier que votre configuraton iptables est correct avec la commande iptables -L -v -n --line-numbers

```
root@debian:~# iptables -L -v --line-numbers
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 machaine   icmp --  any    any     localhost            anywhere

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

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

Chain machaine (1 references)
num   pkts bytes target     prot opt in     out     source               destination
1        0     0 LOG        all  --  any    any     anywhere             anywhere            LOG level warn prefix "PING_BLOCKED
2        0     0 DROP       all  --  any    any     anywhere             anywhere

Chain machaine~ (0 references)
num   pkts bytes target     prot opt in     out     source               destination
```

- Vous pouvez observer que votre chaine est bien mise en place et vos règles aussi sont correctement définis.
- Sur votre machine serveur faites un ping sur vous-même et vous allez observer que les pings seront correctement bloqués

```
root@debian:~# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
```

- Ouvrez aussi un deuxième terminale et verifié le fichier log avec la commande journalctl | grep "Ping_LOCKED" et votre machine vous répondra ceci :

```
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
```

- Explcation du message :
  - mars 11 18:21:21 → Date et heure de l'événement.
  - debian kernel: → Message généré par le noyau Linux.
  - PING_BLOCKED: → Indique que le ping a été bloqué.
  - IN=lo OUT= → L'interface utilisée est lo (boucle locale).
  - MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 → Adresse MAC (pas significative ici car c'est l'interface locale lo).
  - SRC=127.0.0.1 DST=127.0.0.1 → L'adresse source et destination sont 127.0.0.1, donc le ping est effectué en local.
  - LEN=84 → Longueur totale du paquet en octets.
  - TOS=0x00 PREC=0x00 → Paramètres du service du paquet (TOS, priorité).
  - TTL=64 → Durée de vie (Time-To-Live) du paquet.
  - ID=33411 DF → ID du paquet et indicateur "Don't Fragment".
  - PROTO=ICMP → Protocole utilisé est ICMP (Internet Control Message Protocol).
  - TYPE=8 CODE=0 → Type et code ICMP :
  - TYPE=8 signifie une requête "Echo Request" (ping).
  - CODE=0 signifie que c'est une requête normale sans sous-code spécifique.
  - ID=44065 SEQ=57 → Identifiant et numéro de séquence du paquet ICMP.

Travail à faire

Nous allons maintenant effectuer plusieurs manipulations, pour faire ceci nous allons donc devoir remettre les politiques par défaut à 0 avec les commandes :

- iptables -P INPUT ACCEPT
- iptables -P OUTPUT ACCEPT
- iptables -P FORWARD ACCEPT

on vide ensuite les règles déjà appliqué avec les commandes :

- iptables -F
- iptables -X

Vérifie que les commandes ont bien été supprime avec la commande iptbles -L -v -n

```
root@debian:~# iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

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

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

Objectif 1: Empêcher le ping du post serveur sur le poste client

- Nous pouvons remettre en place la règle que nous avions déjà utilisé auparavant qui est :

```
root@debian:~# iptables -A OUTPUT -p icmp --icmp-type echo-request -d 192.168.20.115 -j LOG --log-prefix "PING_BLOCKED: " --log-level
```

Ou bien une règle plus courte et allégé qui est : iptables -A OUTPUT -p icmp --icmp-type echo-request -d 192.168.20.115 -j DROP

```
root@debian:~# iptables -A OUTPUT -p icmp --icmp-type echo-request -d 192.168.20.115 -j DROP
```

Explication de la règle :

- -A OUTPUT : Cela signifie que la règle est ajoutée à la chaîne de sortie (OUTPUT), qui gère les paquets sortants de l'ordinateur.
- -p icmp : Cela indique que la règle s'applique aux paquets du protocole ICMP (Internet Control Message Protocol).
- --icmp-type echo-request : Cela spécifie que la règle concerne uniquement les requêtes "echo", qui sont les requêtes envoyées par la commande ping
- -d 192.168.20.115 : Cela limite l'application de la règle aux paquets destinés à l'adresse IP 192.168.20.115
- -j DROP : Cela indique que les paquets correspondants doivent être abandonnés (DROP), c'est-à-dire qu'ils ne seront pas envoyés

Vérification :

- Sur votre machine serveur faites un ping vers votre machine cliente

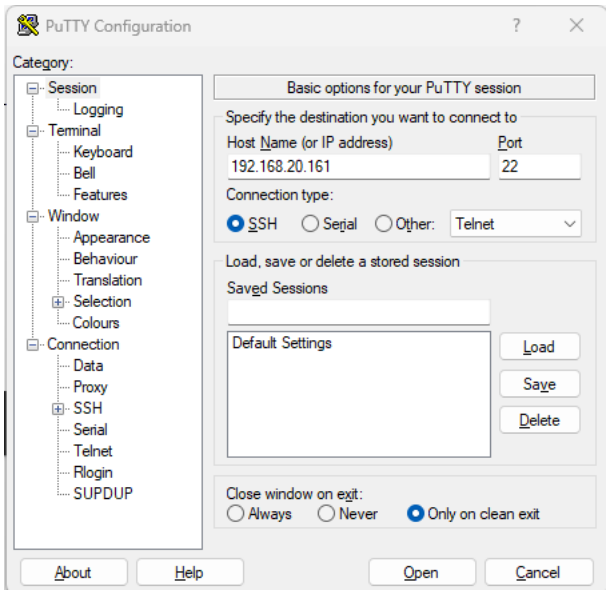
```
root@debian:~# ping 192.168.20.115
PING 192.168.20.115 (192.168.20.115) 56(84) bytes of data.
```

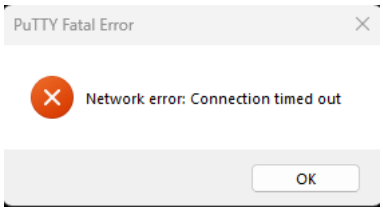
- En même temps sur votre machine serveur ouvrez un autre terminal et entrer la règle : journalctl | grep « Ping_BLOCKED » comme pour la premiere manipulation on peut donc constater que le ping de la machine est bien bloqué

```
mars 11 19:11:28 debian kernel: PING_BLOCKED: IN= OUT=ens18 SRC=192.168.20.161 D
ST=192.168.20.115 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=17329 DF PROTO=ICMP TYPE=8
 CODE=0 ID=17210 SEQ=11
```

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

- Nous allons donc mettre en place la règle qui est : iptables -A INPUT -p tcp --dport 80 -j ACCEPT pour Autoriser seulement le trafic HTTP (port 80)
- Vérification :
  - Sur puTTY





- Utilisé la commande curl sur une machine en mode console :

```
root@debian:~# curl http://192.168.20.161
```

```
      <div class="section_header">
          <div id="docroot"></div>
              Document Roots
      </div>

      <div class="content_section_text">
          <p>
              By default, Debian does not allow access through the web browser to
              <em>any</em> file apart of those located in <tt>/var/www</tt>,
              <a href="http://httpd.apache.org/docs/2.4/mod/mod_userdir.html" rel="nofollow">public_html</a>
              directories (when enabled) and <tt>/usr/share</tt> (for web
              applications). If your site is using a web document root
              located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
              document root directory in <tt>/etc/apache2/apache2.conf</tt>.
          </p>
          <p>
              The default Debian document root is <tt>/var/www/html</tt>. You
              can make your own virtual hosts under /var/www. This is different
              to previous releases which provides better security out of the box.
          </p>
      </div>

      <div class="section_header">
        <div id="bugs"></div>
              Reporting Problems
      </div>
      <div class="content_section_text">
        <p>
              Please use the <tt>reportbug</tt> tool to report bugs in the
              Apache2 package with Debian. However, check <a
              href="http://bugs.debian.org/cgi-bin/pkgreport.cgi?ordering=normal;archive=0;src=apache2;repeatmerged=0"
              rel="nofollow">existing bug reports</a> before reporting a new bug.
        </p>
        <p>
              Please report bugs specific to modules (such as PHP and others)
              to respective packages, not to the web server itself.
        </p>
      </div>




    </div>
  </div>
  <div class="validator">
  </div>
 </body>
</html>
```

- Faites un nmap pour voir les ports ouverts

```
root@debian:~# nmap -p- 192.168.20.161
Starting Nmap 7.93 ( https://nmap.org ) at 2025-03-11 20:45 CET
Nmap scan report for 192.168.20.161
Host is up (0.000016s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 2.24 seconds
```

- On voit le port 22 toujours ouvert meme si on ne pveut pas l'uttiliser (normal on a un serveur ssh sur la machine) donc pour fermé le port on utilise la commande :

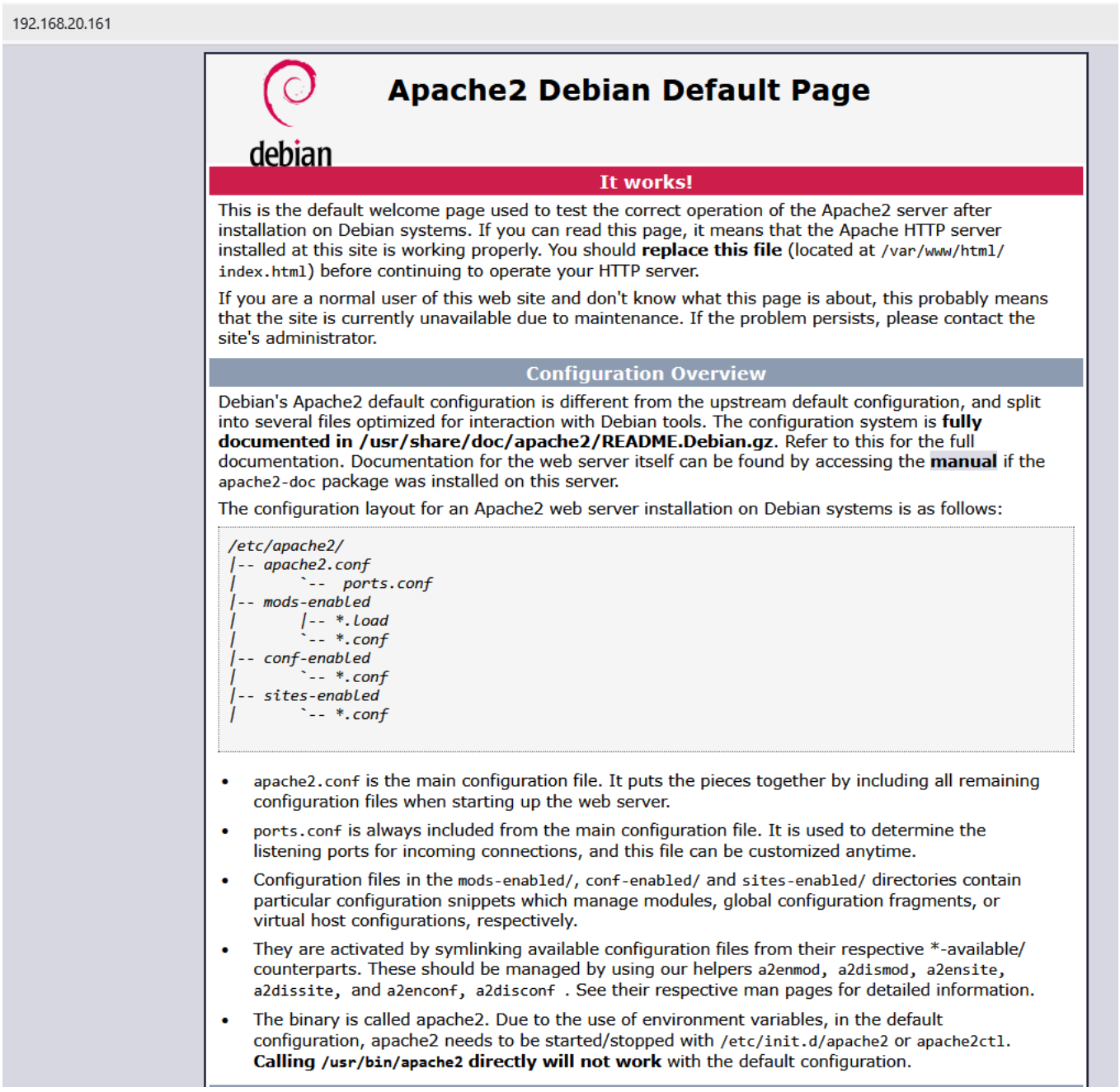
```
iptables -A INPUT -p tcp --dport 22 -j REJECT
```

- Refaites une fois la commande nmap et vous verrez que votre port est bien fermé

```
root@debian:~# nmap -p- 192.168.20.161
Starting Nmap 7.93 ( https://nmap.org ) at 2025-03-11 20:55 CET
Nmap scan report for 192.168.20.161
Host is up (0.00052s latency).
Not shown: 65533 filtered tcp ports (no-response), 1 filtered tcp ports (port-unreach)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: BC:24:11:C4:2C:73 (Unknown)
```

- Accéder a la page web aussi depuis le web

192.168.20.161



Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

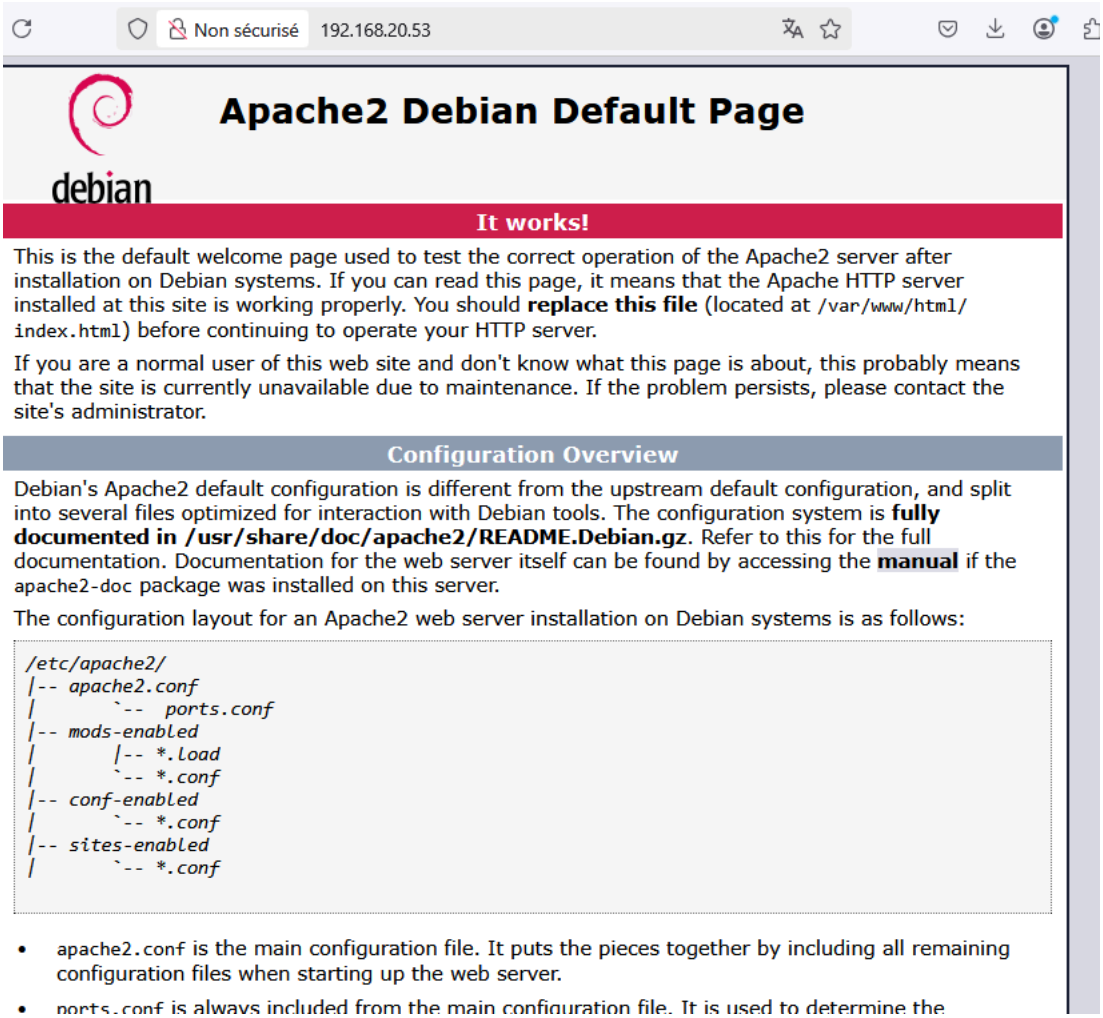
Objectif 3: Interdire l'accès à une seule adresse

- Nous devons faire en sorte d'ajouter un IP dans le fichier nano /etc/network/interfaces et de redémarrer le système réseau aavec la commande systemctl restart networking

```
# The primary network interface
allow-hotplug ens18
iface ens18 inet dhcp

allow-hotplug ens18
iface ens18 inet static
address 192.168.20.53/24
```

```
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:c4:2c:73 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.20.161/24 brd 192.168.20.255 scope global dynamic ens18
       valid_lft 7198sec preferred_lft 7198sec
    inet 192.168.20.53/24 brd 192.168.20.255 scope global secondary ens18
       valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fec4:2c73/64 scope link
       valid_lft forever preferred_lft forever
```

- Essayez d'accéder au serveur depuis la nouvelle IP sur le navigateur web de la machine cliente :



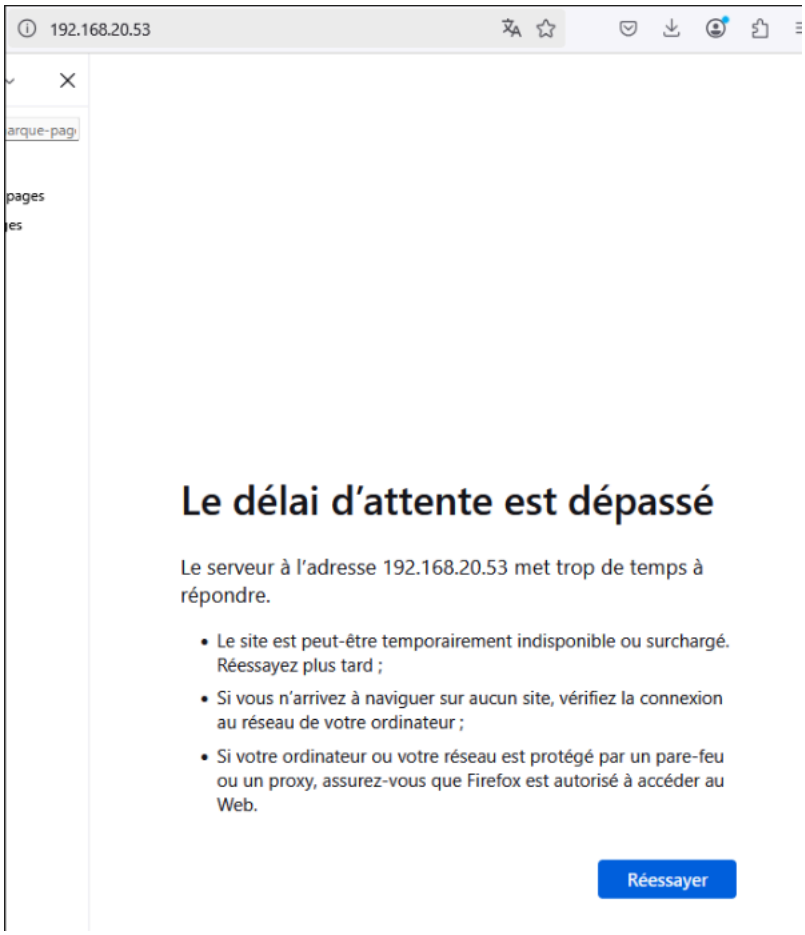
- Nous allons donc maintenant interdire l'accés au site web sur l'adresse IP 192.168.20.53/24
- Nous allons donc utilisé les deux commandes suivantes pour bloquer l'accés http et https:
    - iptables -A INPUT -d 192.168.20.53 -p tcp --dport 80 -j DROP
    - iptables -A INPUT -d 192.168.20.53 -p tcp --dport 443 -j DROP

```
root@debian:~# iptables -A INPUT -d 192.168.20.53 -p tcp --dport 80 -j DROP
```

```
root@debian:~# iptables -A INPUT -d 192.168.20.53 -p tcp --dport 443 -j DROP
```

Vérification :

- Accés depuis le navigateur web :



- Tentative de ping :

```
C:\Users\sio>ping 192.168.20.53

Envoi d'une requête 'Ping'  192.168.20.53 avec 32 octets de données :
Réponse de 192.168.20.53 : octets=32 temps=2 ms TTL=64
Réponse de 192.168.20.53 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.20.53 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.20.53 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.20.53:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 2ms, Moyenne = 1ms
```

- Logs du serveur :

```
root@debian:~# iptables -L -v -n
Chain INPUT (policy ACCEPT 244 packets, 76190 bytes)
 pkts bytes target     prot opt in     out     source               destination
   10   520 DROP       6    --  *      *       0.0.0.0/0            192.168.20.53        tcp dpt:80
    0     0 DROP       6    --  *      *       0.0.0.0/0            192.168.20.53        tcp dpt:443
```

Objectif 4: Refuser toutes connexion telnet

- Règle à mettre en place : iptables -A INPUT -p tcp --dport 23 -j DROP

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

Vérifications :

- Vérifier les logs, et vous observez que la connexion sur le port 23 est réfusé :

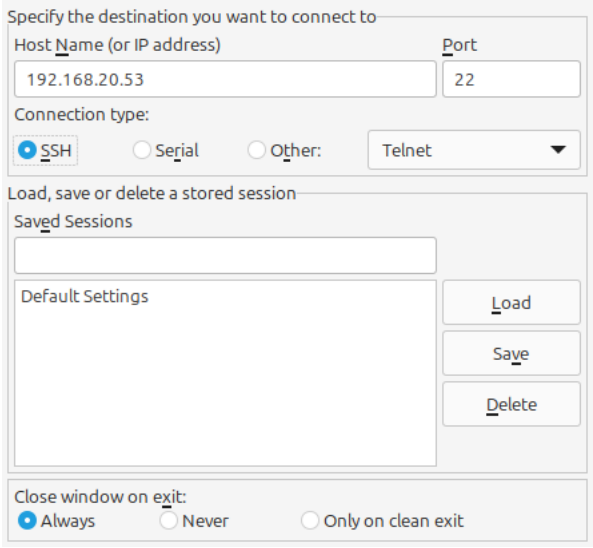
```
root@debian:~# iptables -L -v -n
Chain INPUT (policy ACCEPT 1480 packets, 487K bytes)
 pkts bytes target     prot opt in     out     source               destination
   10   520 DROP       6    --  *      *       0.0.0.0/0            192.168.20.53        tcp dpt:80
    0     0 DROP       6    --  *      *       0.0.0.0/0            192.168.20.53        tcp dpt:443
    0     0 DROP       6    --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:23
```

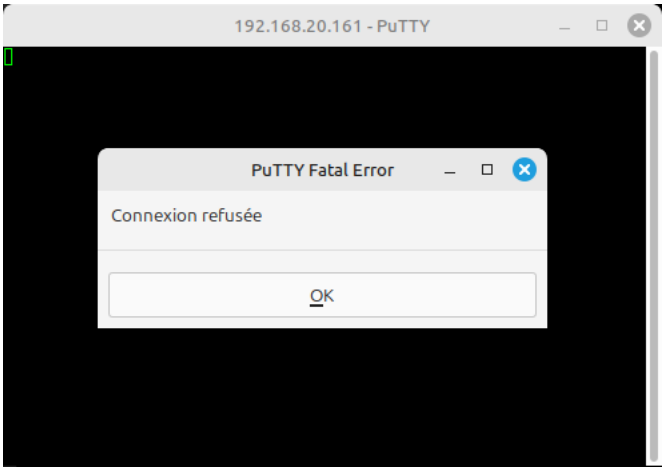
- Faites un namp et vous observerez que la ligne concernant le port 23 est défini comme “filtré”

```
Nmap done: 1 IP address (1 host up) scanned in 104.32 seconds
root@debian:~# nmap -p- 192.168.20.161
Starting Nmap 7.93 ( https://nmap.org ) at 2025-03-11 22:23 CET
Nmap scan report for 192.168.20.161
Host is up (0.00032s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE    SERVICE
22/tcp open     ssh
23/tcp filtered telnet
80/tcp open     http
MAC Address: BC:24:11:C4:2C:73 (Unknown)
```

- Essayer d'accéder a votre machine serveur depuis les ports telnet :

```
Nmap done: 1 IP address (1 host up) scanned in 4.29 seconds
root@debian:~# telnet 192.168.20.161 23
Trying 192.168.20.161...
```





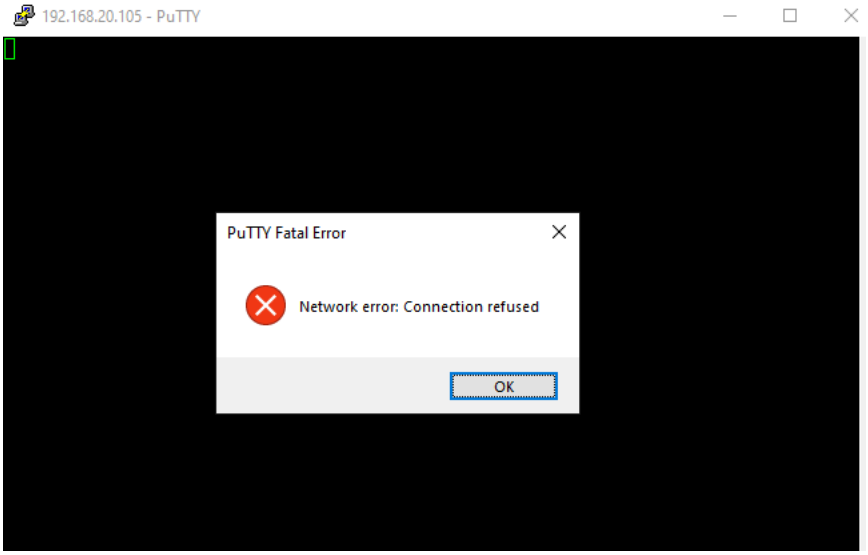
ATTENTION UN CHANGEMENT D'IP DHCP SUR LA MACHIEN CLMIENTE A EU LIEU SUR LA SUITE DU TP LA MACHINE CLIENTE A DONC POUR IP 192.168.20.105 MAINTENANT!

Objectif 5.1: Votre poste client ne peux que consulter votre serveur web:

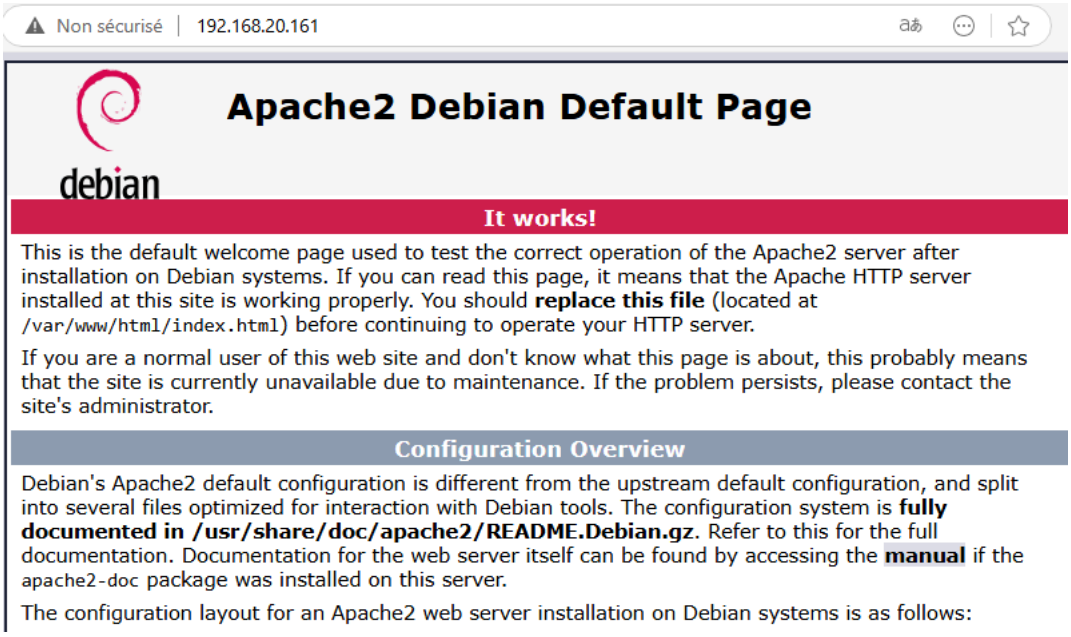
- Regles a mettre en place :
  - iptables -A INPUT -p tcp -s 192.18.20.105 --dport 80 -j ACCEPT (Autorise le trafic HTTP)
  - iptables -A INPUT -s 192.168.20.105 -j DROP (Bloquer tout autre trafic provenant de la machine cliente)
- Vérification:
- aller dans le fichier iptables -L -v -n

```
root@thomas-grzesinki:~# iptables -L -v -n
Chain INPUT (policy DROP 196 packets, 36650 bytes)
 pkts bytes target     prot opt in     out     source               destination
 6218 8304K ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
   10  1312 ACCEPT     0    --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     6    --  *      *       192.168.20.105       0.0.0.0/0           tcp dpt:80
    1   229 DROP       0    --  *      *       192.168.20.105       0.0.0.0/0
```

- Tenter une connexion ssh :



- Entrer l'IP serveur dans un navigateur web :



Objectif 5.2: Votre poste client ne peux pas pinger votre serveur :

- Règles : iptables -I INPUT -s 192.168.20.105 -d 192.16.20.161 -p icmp -j DROP

```
root@debian:~# iptables -I INPUT -s 192.168.20.105 -d 192.168.20.161 -p icmp -j DROP
```

Vérification :

```
C:\Users\sio>ping 192.168.20.161

Envoi d'une requête 'Ping'  192.168.20.161 avec 32 octets de données :
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.

Statistiques Ping pour 192.168.20.161:
    Paquets : envoyés = 4, reçus = 0, perdus = 4 (perte 100%),
```

Objectif 5.3: Votre poste clinet ne peux pas être pingué

Règles : iptables -A INPUT -s 192.168.20.106 -p icmp -j DROP

```
root@debian:~# iptables -A INPUT -s 192.168.20.105 -p icmp -j DROP
```

Vérification :

```
root@deblan:~# ping 192.168.20.105
PING 192.168.20.105 (192.168.20.105) 56(84) bytes of data.
```

Objectif 5.4: Votre serveur web est uniquement serveur web:

- Règles pour autoriser le trafic HTTP et HTTPS et bloquer tout le reste :

```
root@debian:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@debian:~# iptables -A INPUT -p tcp --dport 443 -j ACCEPT
root@debian:~# iptables -A INPUT -j DROP
```

- Vérification :

```
3       20  1460 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
4       30  1560 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
```

Objectif 5.5: Seules les connexions établis sont acceptés

- Règles :

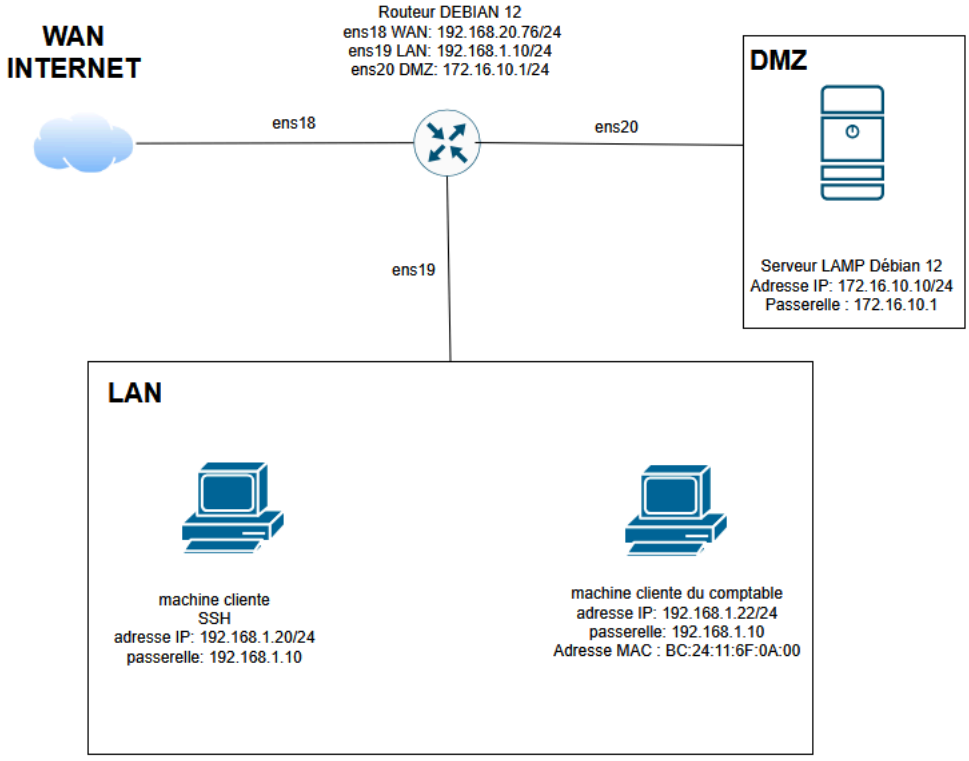
```
root@debian:~# iptables -A INPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

Intégralité des règles du TP :

```
root@debian:~# iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       6    --  *      *       0.0.0.0/0            192.168.20.53        tcp dpt:80
    0     0 DROP       6    --  *      *       0.0.0.0/0            192.168.20.53        tcp dpt:443
    0     0 DROP       6    --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:23
    0     0            0    --  *      *       192.168.0.0          0.0.0.0/0
    0     0 DROP       1    --  *      *       192.168.20.105       192.168.20.161
    0     0 DROP       1    --  *      *       192.168.20.105       0.0.0.0/0
    0     0 ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate ESTABLISHED

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

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 LOG        1    --  *      *       0.0.0.0/0            192.168.20.115       icmptype 8 LOG flags 0 level 4 prefix "PING_BLOCKED: "
    0     0 DROP       1    --  *      *       0.0.0.0/0            192.168.20.115       icmptype 8

Chain machaine (0 references)
 pkts bytes target     prot opt in     out     source               destination

Chain machiane (0 references)
 pkts bytes target     prot opt in     out     source               destination
```

TP2– IPTABLES

Schéma de l’infrastructure



Tuto à suivre : https://www.it-connect.fr/configurer-un-routeur-sous-linux%EF%BB%BF/

Interface réseau:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens18
iface ens18 inet dhcp

allow-hotplug ens19
iface ens19 inet static
        address 192.168.1.10
        netmask 255.255.255.0

allow-hotplug ens20
iface ens20 inet static
        address 172.16.10.1
        netmask 255.255.255.0

post-up iptables-restore </etc/iptables_rules.save
```

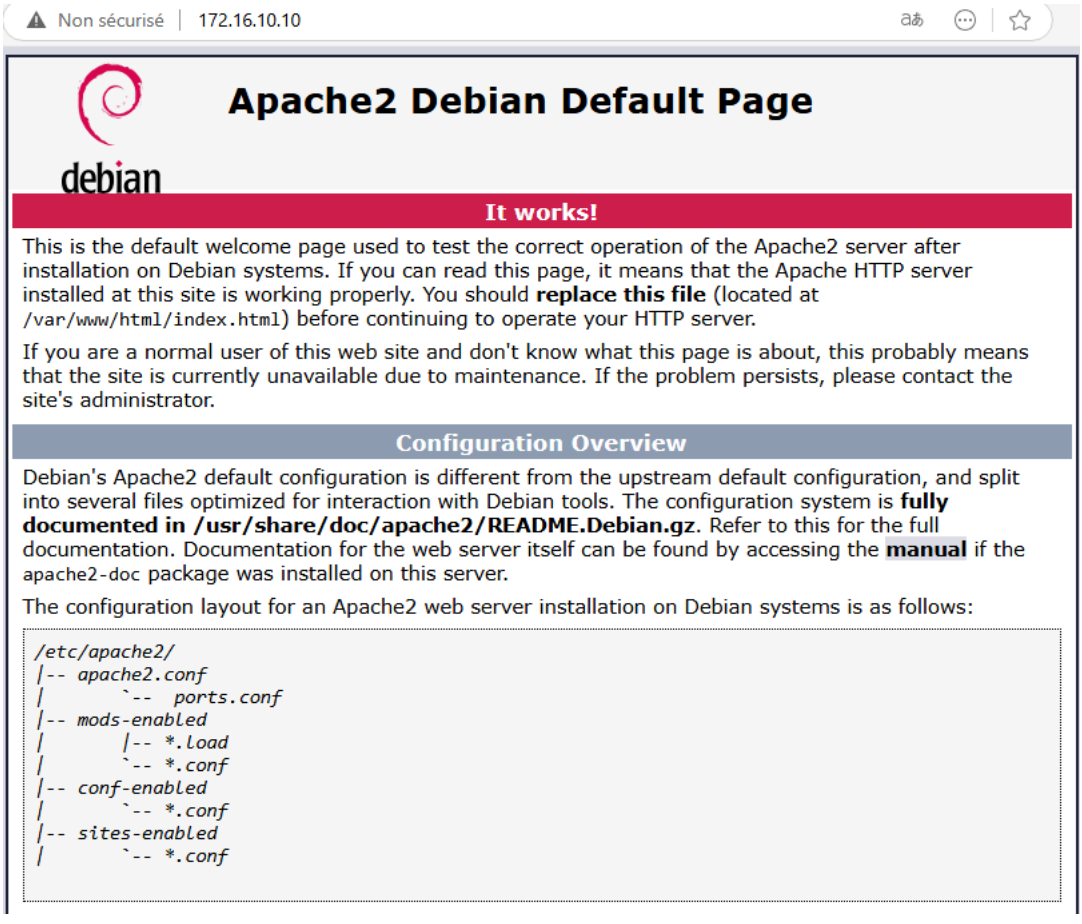
Gestion de la DMZ

- Objectif 1: la DMZ doit être accessible de partout et dont internet et de notre réseau locale
    - Règle : iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 172.16.10.10:80

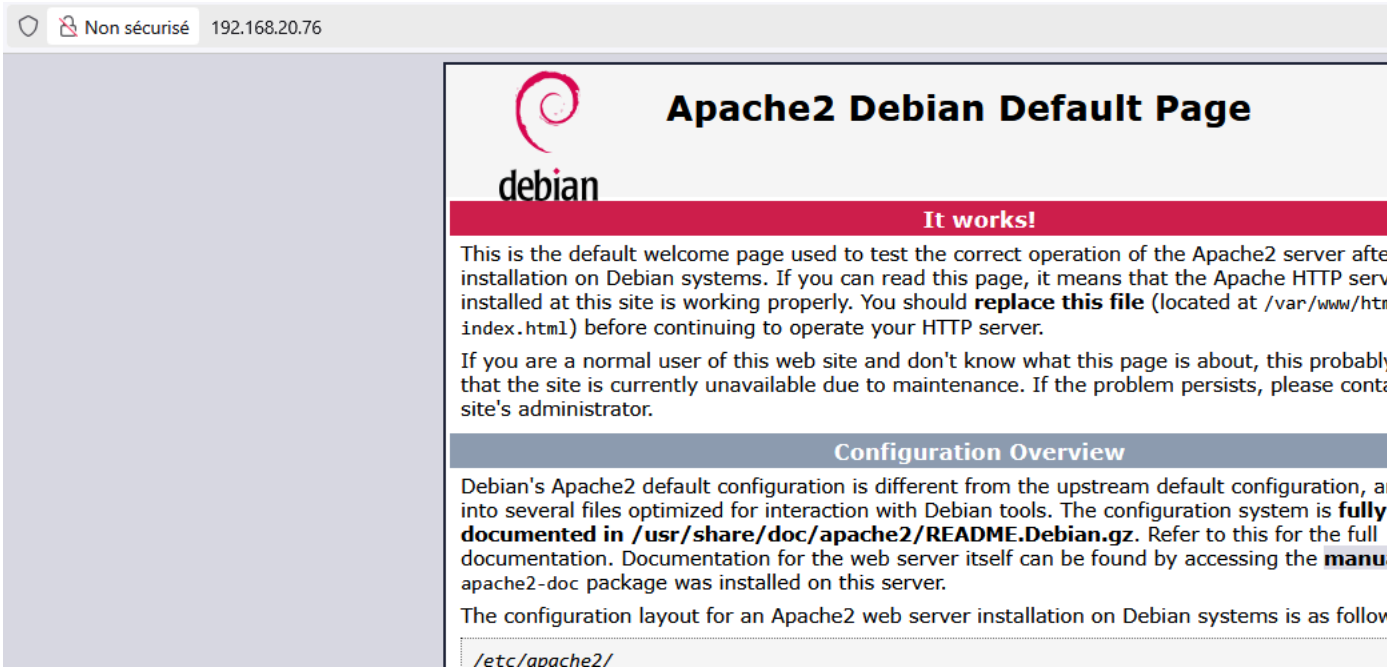
```
root@debian:/var/log# iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 172.16.10.10:80
```

Explication : Cette règle, appliquée dans la table NAT lors du pré-traitement des paquets, redirige tous les paquets TCP destinés au port 80 vers l'adresse 172.16.10.10 sur le même port.

- Vérification :
- LAN :



- WAN :



- Objectif 2 : Empêcher le ping de nos serveurs depuis l'extérieur pour éviter une attaque DDOS
  - Règle : iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

```
root@debian:~# iptables -A INPUT -p icmp --icmp-type echo-request -i ens18 -j DROP
```

Explication : Cette règle ajoute une entrée à la chaîne INPUT pour bloquer (DROP) tous les paquets ICMP de type echo-request, empêchant ainsi les requêtes ping entrantes.

  - Vérification :
  - WAN :

```
root@debian:~# ping 192.168.20.76
PING 192.168.20.76 (192.168.20.76) 56(84) bytes of data.
.
```

- Objectif 3 : Mais le ping depuis le LAN reste possible
  - Règle : iptables -A INPUT -p icmp --icmp-type echo-request -s 192.168.1.0/24 -j ACCEPT

```
root@debian:~# iptables -A INPUT -p icmp --icmp-type echo-request -s 192.168.1.0/24 -j ACCEPT
```

Explication : Cette règle ajoute une entrée à la chaîne INPUT pour autoriser les requêtes ping (ICMP echo-request) provenant du sous-réseau 192.168.1.0/24.

  - Vérification:
    -

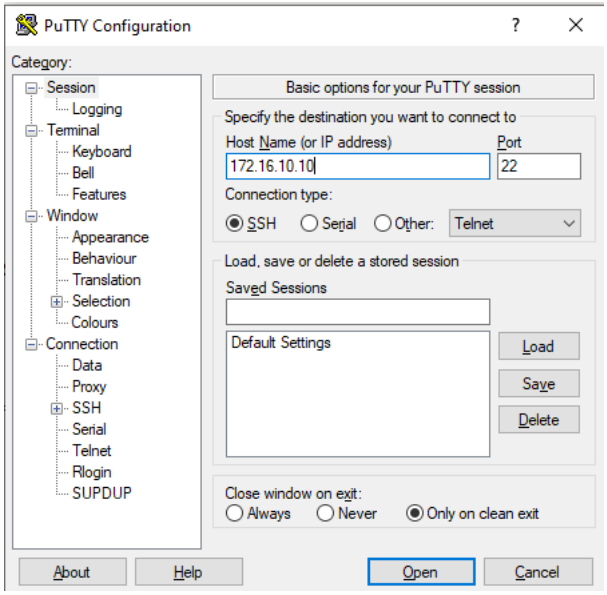
```
C:\Users\sio>ping 172.16.10.10

Envoi d'une requête 'Ping'  172.16.10.10 avec 32 octets de données :
Réponse de 172.16.10.10 : octets=32 temps=3 ms TTL=63
Réponse de 172.16.10.10 : octets=32 temps=2 ms TTL=63
Réponse de 172.16.10.10 : octets=32 temps=2 ms TTL=63
Réponse de 172.16.10.10 : octets=32 temps=2 ms TTL=63

Statistiques Ping pour 172.16.10.10:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 2ms, Maximum = 3ms, Moyenne = 2ms
```

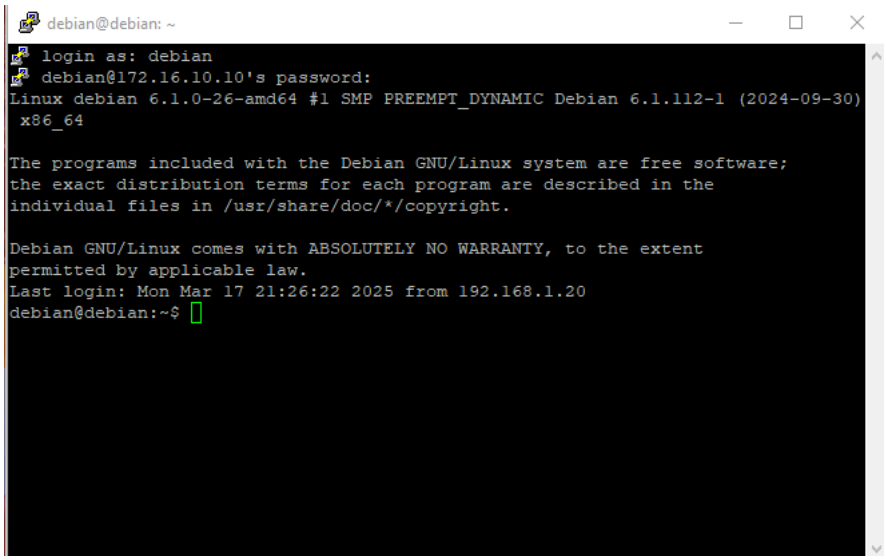
- Objectif 4 : le serveur SSH peut se connecter à la DMZ
  - Règle : iptables -A FORWARD -p tcp -s 192.168.1.10 -d 172.16.10.10 --dport 22 -j ACCEPT

```
root@debian:~# iptables -A FORWARD -p tcp -s 192.168.1.10 -d 172.16.10.10 --dport 22 -j ACCEPT
```

Explication : Cette règle ajoute une entrée à la chaîne FORWARD qui autorise les paquets TCP en provenance de l'adresse 192.168.1.10 vers la destination 172.16.10.10, mais uniquement pour le trafic destiné au port 22 (souvent utilisé pour SSH).

- Vérification:





Gestion de la LAN

Objectif 1 : Autorisez le ping de tous les postes du réseau privé sur l'interface LAN du routeur.

- Règle : iptables -A -INPUT -P icmp --icmp-type echo-request -s 192.168.1.0/21 -d 192.168.1.10 -j ACCEPT

```
root@debian:~# iptables -A INPUT -p icmp --icmp-type echo-request -s 192.168.1.0/24 -d 192.168.1.10 -j ACCEPT
```

  Explication : Cette règle ajoute une entrée dans la chaîne INPUT pour autoriser les paquets ICMP de type echo-request (requêtes ping) provenant du sous-réseau 192.168.1.0/21 et destinés à l'adresse 192.168.1.10.

- Vérification :

```
C:\Users\sio>ping 192.168.1.10

Envoi d'une requête 'Ping'  192.168.1.10 avec 32 octets de données :
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.1.10:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms
```

Objectif 2 : Autorisez le routage des paquets provenant du réseau privé vers la DMZ

- Règle : iptables -A FORWARD -s 192.168.1.0/24 -d 172.16.10.0/24 -j ACCEPT

```
root@debian:~# iptables -A FORWARD -s 192.168.1.0/24 -d 172.16.10.0/24 -j ACCEPT
```

Explication : Cette règle ajoute une entrée à la chaîne FORWARD pour autoriser tous les paquets provenant du sous-réseau 192.168.1.0/24 et destinés au sous-réseau 172.16.10.0/24.

- Vérification :

```
C:\Users\sio>ping 172.16.10.10

Envoi d'une requête 'Ping'  172.16.10.10 avec 32 octets de données :
Réponse de 172.16.10.10 : octets=32 temps=5 ms TTL=63
Réponse de 172.16.10.10 : octets=32 temps=2 ms TTL=63
Réponse de 172.16.10.10 : octets=32 temps=2 ms TTL=63
Réponse de 172.16.10.10 : octets=32 temps=2 ms TTL=63

Statistiques Ping pour 172.16.10.10:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 2ms, Maximum = 5ms, Moyenne = 2ms
```

Objectif 3 : camoufler les adresses Ip des postes du réseau privé qui vont sur internet (voir masquerade)

- Règle : iptables -t nat -A POSTROUTING -o ens18 -s 192.168.1.0/24 -j MASQUERADE

```
root@debian:~# iptables -t nat -A POSTROUTING -o ens18 -s 192.168.1.0/24 -j MASQUERADE
root@debian:~# sysctl -p
net.ipv4.ip_forward = 1
```

Explication : Cette règle, appliquée dans la table NAT, ajoute une entrée à la chaîne POSTROUTING pour modifier l'adresse source des paquets provenant du réseau 192.168.1.0/24 qui sortent par l'interface ens18, en utilisant la cible MASQUERADE pour masquer ces adresses derrière l'adresse IP de l'interface de sortie. sysctl -p permet de redémarrer le service NAT

- Vérification : (On aura donc l'adresse IP WAN qui prendra le relais pour naviguer sur le web, mais impossible de capturer une trame réseaux sur le réseaux de la salle)

Objectif 4 : Le poste comptable ne peut pas aller sur internet

- Règle: iptables -A FORWARD -m mac --mac-source BC:24:11:6F:0A:00 -j DROP

```
root@debian:/var/log# iptables -A FORWARD -m mac --mac-source BC:24:11:6F:0A:00 -j DROP
```

Explication :Cette règle ajoute une entrée à la chaîne FORWARD pour bloquer (DROP) tous les paquets provenant de l'adresse MAC , empêchant ainsi son trafic de transiter par le pare-feu pour anviguer sur internet.

- Vérification:

```
C:\Users\sio>ping 8.8.8.8

Envoi d'une requête 'Ping'  8.8.8.8 avec 32 octets de données :
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.

Statistiques Ping pour 8.8.8.8:
    Paquets : envoyés = 4, reçus = 0, perdus = 4 (perte 100%),

C:\Users\sio>ping 192.168.1.10

Envoi d'une requête 'Ping'  192.168.1.10 avec 32 octets de données :
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.10 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.1.10:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms
```

Objectif 5 : on n'acceptera que les connexions établies

- Règle : iptables -A INPUT -m conntrack --ctsate ESTABLISHED,RELATED -j ACCEPT

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

Explication : Cette règle ajoute une entrée à la chaîne INPUT pour autoriser tous les paquets appartenant à une connexion déjà établie ou liée à une connexion existante en utilisant le module conntrack.

Objectif 6 : Les paquets rejetés seront enregistrés dans le journal /var/log/syslog

- Règle : iptables -A INPUT -i ens18 -j LOG --log-prefix "REFUS: " --log-level 4

```
root@debian:/var/log# iptables -A INPUT -i ens18 -j LOG --log-prefix "REFUS: " --log-level 4
```

Explication : Cette règle iptables ajoute une entrée à la chaîne INPUT pour enregistrer en log, avec le préfixe “REFUS: ” et un niveau de log 4, tous les paquets entrants sur l’interface ens18.

- Vérification : Entrer la commande tail -f /var/log/syslog

```
root@debian:/var/log# tail -f /var/log/syslog
2025-03-22T14:49:30.844080+01:00 debian kernel: [351409.172348] REFUS: IN=ens18 OUT= MAC=ff:ff:ff:ff:ff:ff:78:8a:20:53:51:d8:08:00
5.255.255 LEN=275 TOS=0x00 PREC=0x00 TTL=64 ID=62435 DF PROTO=UDP SPT=52012 DPT=10001 LEN=255
```

```
2025-03-22T12:14:56.015796+01:00 debian kernel: [342137.445604] REFUS: IN=ens18 OUT= MAC=ff:ff:ff:ff:ff:ff:78:8a:20:53:51:d8:08:00 SRC=
5.255.255 LEN=275 TOS=0x00 PREC=0x00 TTL=64 ID=21264 DF PROTO=UDP SPT=56167 DPT=10001 LEN=255
```

Intégralité des règles du TP-2 :

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    4   240 ACCEPT     1    --  *      *       192.168.1.0/24       192.168.1.10         icmptype 8
    0     0 ACCEPT     1    --  *      *       0.0.0.0/0            192.168.1.0/24       icmptype 8
    0     0 DROP       1    --  ens187 *       0.0.0.0/0            0.0.0.0/0            icmptype 8
    5   380 ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
   53 14315 LOG        0    --  ens18  *       0.0.0.0/0            0.0.0.0/0            LOG flags 0 level 4 prefix "REFUS: "

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  673 66690 ACCEPT     0    --  *      *       192.168.1.0/24       172.16.10.0/24
    0     0 ACCEPT     6    --  *      *       192.168.1.10         172.16.10.10         tcp dpt:22
 1657  133K DROP       0    --  *      *       0.0.0.0/0            0.0.0.0/0            MAC bc:24:11:6f:0a:00

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