TP 3

Administration système

Dardor Rochdi

Exercice 3.1 :

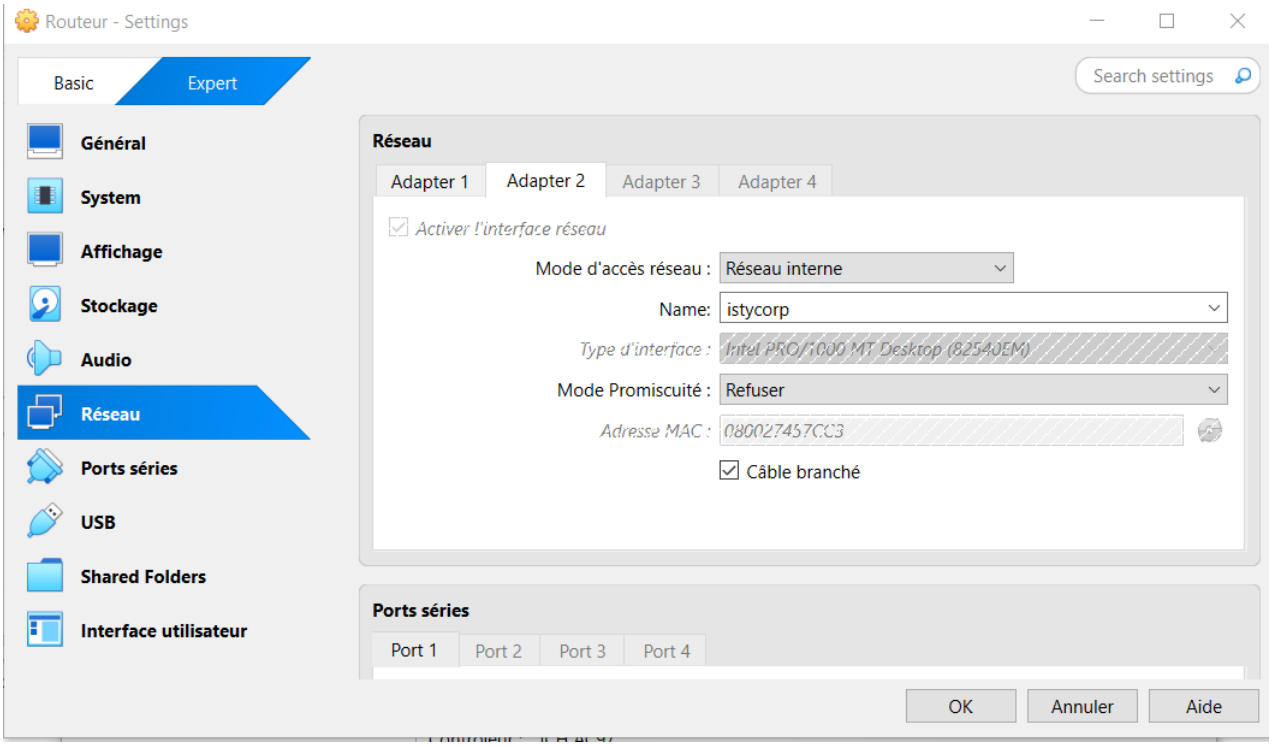
L’objectif de cet exercice est de mettre en place deux machines virtuelles pour simuler une infrastructure réseau. Une machine jouera le rôle de routeur, et l’autre celui d’un client connecté via le réseau interne.

Détails de la configuration :

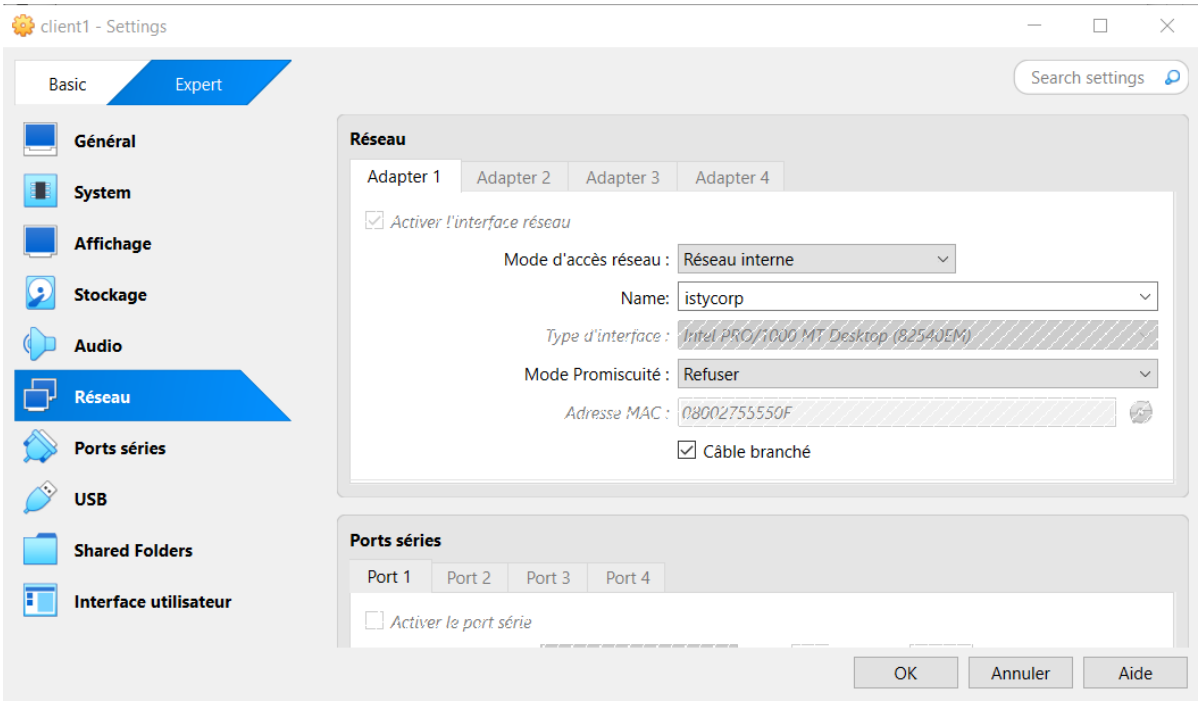
1. **Routeur** :
   - Deux interfaces réseau ont été configurées :
     1. **Interface NAT** : Connectée à Internet via l’hôte physique, permettant au routeur de servir de passerelle pour le client.
     2. **Interface interne** : Connectée à un réseau privé (nommé **istycorp**) pour assurer la communication avec Client1.
   - Système d'exploitation : Debian GNU/Linux.
2. **Client1** :
   - Une seule interface réseau connectée au réseau interne (**istycorp**).
   - Le client communique uniquement avec le routeur pour accéder aux ressources du réseau ou à Internet.
   - Système d'exploitation : Debian GNU/Linux.

Configuration réseau des machines dans VirtualBox :

- Pour le routeur :
  - **Adaptateur 1** : Configuré en NAT.
  - **Adaptateur 2** : Configuré en réseau interne nommé **istycorp**.



- Pour Client1 :
    - **Adaptateur 1** : Configuré en réseau interne **istycorp**.



EXERCICE 3.2 :

Connecter directement les machines clients à Internet présente plusieurs inconvénients majeurs, tant sur le plan de la sécurité que de la gestion réseau. Voici trois raisons principales pour lesquelles cette approche est généralement évitée :

1. Sécurité: Lorsque vous connectez des machines directement à Internet, elles deviennent directement exposées aux menaces en ligne telles que les attaques par force brute, les scans de ports, les attaques DDoS, etc. Un réseau local agit comme une couche de protection en isolant les machines internes de l'exposition directe à Internet. Cela permet de mettre en place des pare-feu et d'autres dispositifs de sécurité pour contrôler et surveiller le trafic entrant et sortant.

2. Gestion du trafic: Un réseau local permet de gérer efficacement le trafic entre les machines internes et d'établir des règles de communication. Cela facilite la mise en place de services locaux tels que le partage de fichiers, l'impression, le partage de ressources, etc. En centralisant ces services au sein du réseau local, on peut mieux contrôler les flux de données et améliorer la performance du réseau.

3. Adressage privé: Les adresses IP privées (par exemple, celles dans les plages 192.168.x.x) sont utilisées dans les réseaux locaux. Ces adresses ne sont pas routables sur Internet, ce qui signifie qu'elles ne sont pas directement accessibles depuis l'extérieur du réseau local. Cela ajoute une couche supplémentaire de sécurité en

obscurcissant les adresses IP réelles des machines internes pour les utilisateurs et les appareils externes.

EXERCICE 3.3 :

Rôle du serveur DHCP

Le serveur DHCP (Dynamic Host Configuration Protocol) automatise la configuration réseau des machines clientes. Il attribue dynamiquement les informations nécessaires pour qu'une machine puisse communiquer sur le réseau. Ces informations incluent notamment :

- L’adresse IP.
- Le masque de sous-réseau.
- La passerelle par défaut.

Les serveurs DNS. :

L'utilisation d'un serveur DHCP simplifie la gestion réseau, surtout dans les environnements avec de nombreux clients. Elle évite les erreurs liées à une configuration manuelle (comme les conflits d’adresses IP) et permet de centraliser les paramètres réseau.

EXERCICE 3.4 :

Pour faciliter l'administration à distance du routeur, un serveur SSH a été installé sur celui-ci. Le paquet **OpenSSH** a été installé à l'aide de la **commande apt install openssh-server.** Une fois l'installation terminée, le service SSH a été démarré et activé pour s'assurer qu'il fonctionne après un redémarrage. La commande **systemctl status ssh** a permis de vérifier que le service était actif.

```
root@vbox:/home/vboxuser# sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-12-25 01:50:09 +01; 10s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3414 (sshd)
      Tasks: 1 (limit: 2284)
     Memory: 1.5M
        CPU: 28ms
     CGroup: /system.slice/ssh.service
             └─3414 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 25 01:50:09 vbox systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Dec 25 01:50:09 vbox sshd[3414]: Server listening on 0.0.0.0 port 22.
Dec 25 01:50:09 vbox sshd[3414]: Server listening on :: port 22.
Dec 25 01:50:09 vbox systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

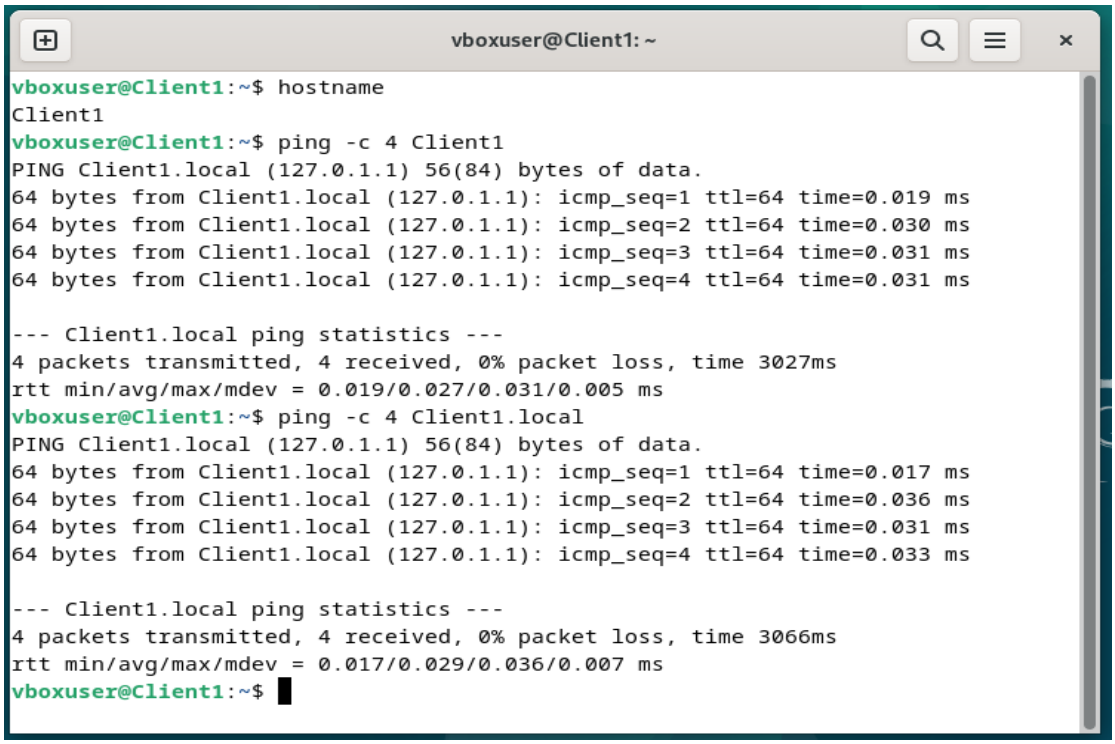
Afin d'accéder au routeur depuis la machine hôte, une redirection de port a été configurée dans VirtualBox. Dans les paramètres de la machine virtuelle routeur, une règle de redirection a été ajoutée dans l'onglet réseau pour l'adaptateur NAT. Cette règle redirige le port 2222 de l'hôte vers le port 22 de la machine routeur.

```
root@vbox:/home/vboxuser# ssh root@192.168.0.1
The authenticity of host '192.168.0.1 (192.168.0.1)' can't be established.
ED25519 key fingerprint is SHA256:EL3cVOOijQulgqnQcoXAUa47Q57QzdkDBxNZMzmKdBQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.0.1' (ED25519) to the list of known hosts.
root@192.168.0.1's password:
Linux vbox 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) 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.
root@vbox:~#
```

Enfin, la connexion SSH au routeur a été testée avec succès depuis la machine hôte en utilisant la commande ssh -p 2222 root@192.168.0.1. Cette configuration permet d'administrer efficacement le routeur à distance, sans nécessiter un accès direct à l'interface VirtualBox.

EXERCICE 3.5 :

Pour personnaliser la configuration du client, le nom d'hôte de la machine a été modifié afin de faciliter son identification dans le réseau. Cette opération a été réalisée en modifiant le fichier /etc/hostname, où le nouveau nom d'hôte, par exemple Client1, a été enregistré. De plus, le fichier /etc/hosts a été ajusté pour associer ce nom d'hôte à l'adresse locale 127.0.1.1, garantissant une résolution locale correcte. Après ces changements, un redémarrage de la machine ou l'utilisation de la commande **hostnamectl set-hostname** Client1 a permis d'appliquer la modification.



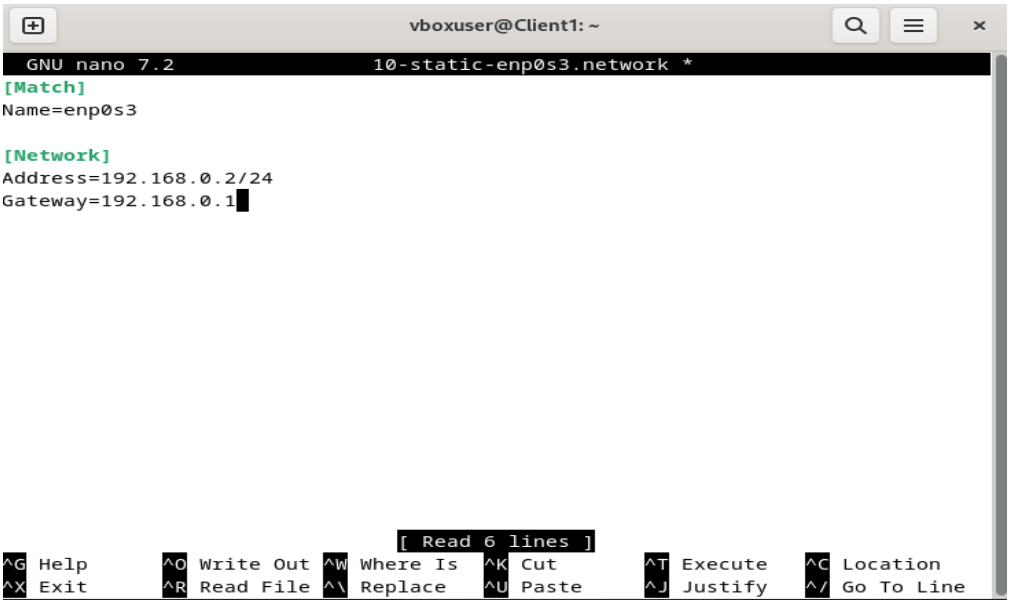
La vérification du domaine local a ensuite été effectuée en inspectant le fichier /etc/resolv.conf. Ce fichier a confirmé l'utilisation d'un domaine local, par exemple istycorp.fr, indiqué par la directive search. Ces configurations assurent une meilleure intégration du client dans le réseau, facilitant ainsi les communications et la gestion des ressources partagées.

Sur le router :

```
root@vbox:/etc/systemd/network# ping -c 4 Client1
PING client1.istycorp.fr (192.168.0.3) 56(84) bytes of data.
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=1 ttl=64 time=0.305 ms
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=2 ttl=64 time=0.508 ms
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=3 ttl=64 time=0.525 ms
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=4 ttl=64 time=0.508 ms

--- client1.istycorp.fr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 0.305/0.461/0.525/0.090 ms
root@vbox:/etc/systemd/network#
```

Exercice 3.6 :

Pour configurer manuellement les adresses IP du routeur et du client, la méthode utilisée repose sur **systemd-networkd**. Sur le routeur, l'adresse IP **192.168.0.1** a été assignée à l'interface interne en modifiant le fichier de configuration réseau /etc/systemd/network/10-enp0s8.network. De manière similaire, sur le client, l'adresse IP **192.168.0.2** a été attribuée en modifiant le fichier /etc/systemd/network/10-enp0s3.network. Après chaque modification, le service réseau a été redémarré à l'aide de la commande systemctl restart systemd-networkd, appliquant ainsi les nouvelles configurations.



Une fois les adresses IP configurées, des tests de connectivité ont été réalisés à l'aide de la commande ping. Depuis le client, un ping 192.168.0.1 a confirmé que le routeur était accessible, et depuis le routeur, un ping 192.168.0.2 a validé que la communication avec le client était également fonctionnelle. Enfin, l'analyse des tables

de routage à l’aide de la commande ip route a révélé que chaque machine disposait d’une route pour le sous-réseau **192.168.0.0/24**, permettant une communication locale. Cependant, aucune passerelle par défaut n’ayant été configurée, les communications restent limitées au réseau interne.

Cette configuration a permis de valider le fonctionnement du réseau interne entre le client et le routeur, préparant ainsi les étapes suivantes pour un accès externe.

```
root@vbox:/etc/systemd/network# ip route
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s8 scope link metric 1000
192.168.0.0/24 dev enp0s8 proto kernel scope link src 192.168.0.1
```

default via 10.0.2.2 dev enp0s3 :

- Cette ligne définit la passerelle par défaut.
- Tout le trafic non local (adresse IP hors des sous-réseaux configurés) est envoyé vers l'adresse **10.0.2.2** via l'interface **enp0s3** (NAT).
- Cela permet au routeur d'accéder à Internet.

10.0.2.0/24 dev enp0s3 :

- Cette route indique que le sous-réseau **10.0.2.0/24** (réseau NAT) est directement accessible via l'interface **enp0s3**.

192.168.0.0/24 dev enp0s8 :

- Cette route indique que le sous-réseau interne **192.168.0.0/24** est accessible via l'interface **enp0s8**.
- Elle est utilisée pour communiquer avec les machines internes comme Client1.

```
root@Client1:/etc/systemd/network# ip route
default via 192.168.0.1 dev enp0s3
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.3
root@Client1:/etc/systemd/network# sudo nano /etc/network/interfaces
```

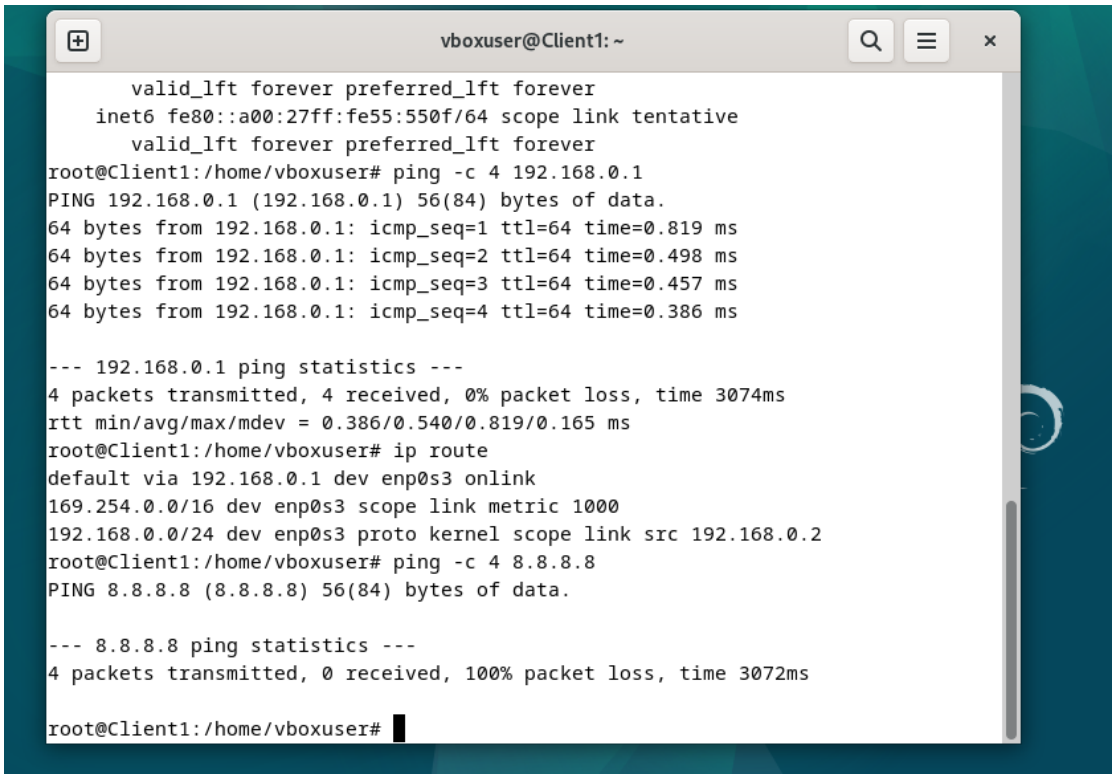
default via 192.168.0.1 dev enp0s3 :

- Cette ligne définit la passerelle par défaut.

Tout le trafic non local (adresse IP hors du réseau interne) est envoyé au routeur via l'adresse **192.168.0.1** sur l'interface **enp0s3**.

192.168.0.0/24 dev enp0s3 :

- Cette route indique que le sous-réseau interne **192.168.0.0/24** est directement accessible via l'interface **enp0s3**.
- Utilisée pour communiquer avec le routeur ou d'autres machines internes.




```
root@vbox:/home/vboxuser# ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=255 time=51.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=255 time=42.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=255 time=42.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=255 time=43.0 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 42.351/44.817/51.005/3.581 ms
root@vbox:/home/vboxuser#
```

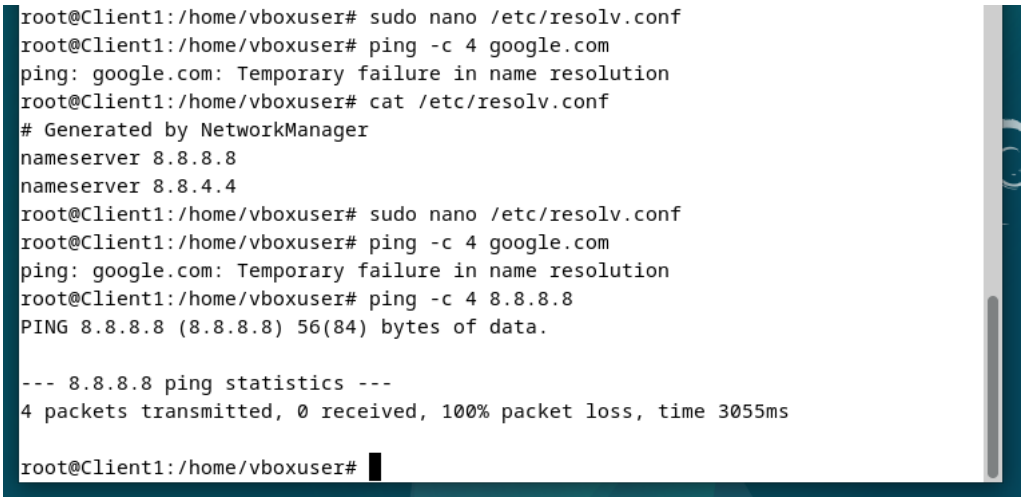
EXERCICE 3.7 :

Pour permettre au client de résoudre des noms de domaine comme le routeur, la configuration DNS du client a été alignée sur celle du routeur. Cela a été réalisé en modifiant le fichier /etc/resolv.conf sur le client. Le contenu du fichier a été ajusté pour pointer vers le serveur DNS externe utilisé par le routeur (par exemple, 10.0.2.3) : nameserver 10.0.2.3

Cette configuration a permis au client d'envoyer ses requêtes DNS au serveur externe via le routeur.

Test de résolution de nom et accès à Internet

1. **Test de résolution de noms** : La commande `ping google.com` a été exécutée sur le client pour vérifier si les noms de domaine étaient correctement résolus.
    - Résultat : La résolution de nom a échoué avec une erreur indiquant un échec temporaire de résolution DNS.
2. **Test d'accès direct à Internet** : La connectivité Internet a été testée en pingant une adresse IP directement (par exemple, `8.8.8.8`) : `ping -c 4 8.8.8.8`
    - Résultat : Le test a échoué, confirmant que le client ne pouvait pas se connecter à Internet.



Pourquoi cela ne fonctionne-t-il pas ?

Le problème réside dans l'absence de configuration de la **passerelle par défaut** sur le client. Bien que le DNS soit correctement configuré, le client ne peut pas envoyer ses paquets en dehors du réseau interne (192.168.0.0/24) sans une passerelle définie pour atteindre les réseaux externes. Le routeur, qui agit comme passerelle, n'a pas été spécifié dans les paramètres réseau du client.

Mise en place du NAT

EXERCICE 3.8 :

Le Network Address Translation (NAT) est une technique essentielle dans les réseaux informatiques, visant à surmonter la limitation d'adresses IP publiques en attribuant des adresses privées aux dispositifs internes et en les faisant apparaître comme une seule adresse IP publique lors de la communication avec Internet. Il offre une économie d'adresses IP publiques, renforce la sécurité en masquant la topologie interne du réseau, et prolonge l'utilisation d'IPv4 malgré la rareté des adresses.

Toutefois Avec l'avènement d'IPv6 et son espace d'adressage considérablement étendu (128 bits par rapport aux 32 bits d'IPv4), le nombre potentiel d'adresses IP uniques devient pratiquement illimité. Contrairement à IPv4, où la rareté des adresses IP publiques a conduit à la mise en œuvre généralisée du NAT, IPv6 offre suffisamment d'adresses pour attribuer une adresse publique unique à chaque dispositif connecté à Internet. Cette abondance d'adresses IPv6 élimine le besoin fondamental du NAT en tant que mécanisme d'économie d'adresses, car chaque dispositif peut avoir une identité unique sans la nécessité de masquer les adresses privées derrière une adresse IP publique partagée. Cependant, bien que le NAT puisse être moins nécessaire en termes de pénurie d'adresses, il peut toujours être utilisé pour des considérations de sécurité en masquant la topologie interne du réseau, même dans un environnement IPv6. Ainsi, l'adoption généralisée d'IPv6 offre une solution élégante au problème de rareté d'adresses qui a historiquement justifié l'utilisation répandue du NAT. Cette transition vers IPv6 permet une allocation d'adresses plus directe, simplifiant la gestion du réseau et réduisant la complexité associée au NAT.

EXERCICE 3.9 :

Pour permettre au routeur de transférer les paquets entre ses interfaces réseau et d’assurer la communication entre le réseau interne et l’extérieur, le routage des

```
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
```

paquets a été activé. Cette configuration a été rendue permanente en modifiant le fichier /etc/sysctl.conf et en ajoutant ou décommentant la ligne net.ipv4.ip_forward = 1. Après modification, les paramètres ont été rechargés à l'aide de la commande **sysctl -p**. Pour activer immédiatement le routage sans redémarrage, la commande sysctl -w **net.ipv4.ip_forward=1** a été utilisée, ou alternativement,

l'instruction echo 1 > /proc/sys/net/ipv4/ip_forward. Ces commandes permettent d'appliquer temporairement le routage pour la session en cours. Cette étape est cruciale pour établir une passerelle fonctionnelle entre le réseau interne et Internet, permettant au routeur de jouer pleinement son rôle dans la transmission des paquets.

```
root@vbox:/home/vboxuser# sudo nano /etc/sysctl.conf
root@vbox:/home/vboxuser# sudo sysctl -p
net.ipv4.ip_forward = 1
root@vbox:/home/vboxuser# sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@vbox:/home/vboxuser# cat /proc/sys/net/ipv4/ip_forward
1
root@vbox:/home/vboxuser# sudo sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
root@vbox:/home/vboxuser#
```

EXERCICE 3.10 :

Pour permettre aux machines du réseau interne d'accéder à Internet via le routeur, le support du NAT (Network Address Translation) a été activé. Une règle a été ajoutée à la table POSTROUTING d'iptables pour masquer les adresses IP privées du réseau interne et les traduire en l'adresse IP publique du routeur. La commande utilisée **était iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o enp0s3 -j MASQUERADE**, où 192.168.0.0/24 représente le réseau interne, et enp0s3 est l'interface externe connectée à Internet. Cette configuration assure que tous les paquets provenant du réseau interne sortent avec l'adresse IP publique du routeur.

```
root@vbox:/home/vboxuser# sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o enp0s3 -j MASQUERADE
root@vbox:/home/vboxuser# sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (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

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  0    --  *      enp0s3  192.168.0.0/24       0.0.0.0/0
root@vbox:/home/vboxuser#
```

Pour rendre cette règle persistante, le paquet iptables-persistent a été installé, puis les règles ont été sauvegardées à l'aide de la commande iptables-save > /etc/rules.ipv4. Ainsi, la configuration NAT est automatiquement restaurée après chaque

redémarrage. Cette étape garantit une communication efficace entre le réseau interne et Internet.

```
root@vbox:/home/vboxuser# sudo iptables-save > /etc/rules.ipv4
root@vbox:/home/vboxuser# sudo netfilter-persistent save
sudo netfilter-persistent reload
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
root@vbox:/home/vboxuser# sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (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

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  0    --  *      enp0s3  192.168.0.0/24       0.0.0.0/0
root@vbox:/home/vboxuser# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.8.9 (nf_tables) on Wed Dec 25 03:39:39 2024
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 192.168.0.0/24 -o enp0s3 -j MASQUERADE
COMMIT
# Completed on Wed Dec 25 03:39:39 2024
root@vbox:/home/vboxuser#
```

```
vboxuser@vbox:~$ sudo iptables -t nat -L -n -v
[sudo] password for vboxuser:
vboxuser is not in the sudoers file.
vboxuser@vbox:~$ su
Password:
root@vbox:/home/vboxuser# sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 1 packets, 576 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 1 packets, 576 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 17 packets, 2559 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 17 packets, 2559 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  0    --  *      enp0s3  192.168.0.0/24       0.0.0.0/0
root@vbox:/home/vboxuser#
```

EXERCICE 3.11 :

Pour permettre au client d'accéder à Internet via le routeur, une passerelle par défaut a été configurée. Cette opération a été réalisée en ajoutant une route par défaut pointant vers le routeur avec la commande **ip route add default via 192.168.0.1 dev enp0s3**, où 192.168.0.1 est l'adresse IP du routeur, et enp0s3 est l'interface réseau du client.

```
root@Client1:/home/vboxuser# ip route
default via 192.168.0.1 dev enp0s3 onlink
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.2
```

Cette configuration permet au client d'envoyer tous les paquets destinés à des réseaux externes vers le routeur, qui agit comme passerelle. La table de routage a été vérifiée avec la commande **ip route**, confirmant l'ajout de la route par défaut. Des tests ont ensuite été effectués, notamment un ping vers une adresse IP externe (8.8.8.8) et un ping vers un domaine (google.com), validant ainsi l'accès à Internet. Cette étape garantit que le client peut communiquer efficacement avec des réseaux externes via le routeur.

```
root@Client1:/home/vboxuser# ip route
default via 192.168.0.1 dev enp0s3 onlink
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.2
root@Client1:/home/vboxuser# ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=254 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=254 time=43.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=254 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=254 time=43.0 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 42.988/43.715/44.363/0.488 ms
root@Client1:/home/vboxuser# ping -c 4 google.com
PING google.com (142.250.185.14) 56(84) bytes of data.
64 bytes from mad41s11-in-f14.1e100.net (142.250.185.14): icmp_seq=1 ttl=254 time=43.7 ms
64 bytes from mad41s11-in-f14.1e100.net (142.250.185.14): icmp_seq=2 ttl=254 time=42.6 ms
64 bytes from mad41s11-in-f14.1e100.net (142.250.185.14): icmp_seq=3 ttl=254 time=41.8 ms
64 bytes from mad41s11-in-f14.1e100.net (142.250.185.14): icmp_seq=4 ttl=254 time=42.0 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 41.832/42.527/43.678/0.730 ms
```

EXERCICE 3.12 :

Pour tester la connexion Internet du client et analyser le routage des paquets, deux outils ont été utilisés : **traceroute** sur le client pour visualiser les étapes de transmission des paquets, et **tcpdump** sur le routeur pour observer le trafic réseau en temps réel.

1.Test avec traceroute sur le client :

- Sur le client, l'outil traceroute a été utilisé pour suivre le chemin des paquets envoyés vers un domaine externe, tel que Google :

```
root@Client1:/home/vboxuser# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  0.495 ms  0.454 ms  0.438 ms
 2  10.0.2.2 (10.0.2.2)  0.607 ms  0.590 ms  0.508 ms
 3  * * *
```

Cette commande a affiché les différents sauts effectués par les paquets depuis le client jusqu'à la destination. Le premier saut correspondait à l'adresse IP du routeur (**192.168.0.1**), confirmant que le routeur jouait bien son rôle de passerelle. Les sauts suivants montraient le chemin emprunté sur Internet jusqu'au serveur final.

2. Observation avec tcpdump sur le routeur

- Pour analyser le trafic réseau en temps réel, l'outil **tcpdump** a été exécuté sur le routeur. Cela a permis de capturer et d'afficher les paquets traversant le routeur, notamment ceux provenant du client :

```
root@vbox:/home/vboxuser# sudo tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
03:58:49.982548 IP vbox > dns.google: ICMP echo request, id 3528, seq 1, length 64
03:58:49.993010 IP vbox.52391 > 10.0.2.3.domain: 25419+ PTR? 8.8.8.8.in-addr.arpa. (38)
03:58:50.025456 IP dns.google > vbox: ICMP echo reply, id 3528, seq 1, length 64
03:58:50.034622 IP 10.0.2.3.domain > vbox.52391: 25419 1/0/0 PTR dns.google. (62)
03:58:50.095671 IP vbox.47730 > 10.0.2.3.domain: 44008+ PTR? 3.2.0.10.in-addr.arpa. (39)
03:58:50.099465 IP 10.0.2.3.domain > vbox.47730: 44008 NXDomain* 0/0/0 (39)
03:58:50.984597 IP vbox > dns.google: ICMP echo request, id 3528, seq 2, length 64
03:58:51.027520 IP dns.google > vbox: ICMP echo reply, id 3528, seq 2, length 64
03:58:51.986501 IP vbox > dns.google: ICMP echo request, id 3528, seq 3, length 64
03:58:52.029586 IP dns.google > vbox: ICMP echo reply, id 3528, seq 3, length 64
03:58:52.987874 IP vbox > dns.google: ICMP echo request, id 3528, seq 4, length 64
03:58:53.030807 IP dns.google > vbox: ICMP echo reply, id 3528, seq 4, length 64
03:58:55.199502 ARP, Request who-has 10.0.2.3 tell vbox, length 28
03:58:55.199510 ARP, Request who-has _gateway tell vbox, length 28
03:58:55.200065 ARP, Reply 10.0.2.3 is-at 52:55:0a:00:02:03 (oui Unknown), length 50
03:58:55.200072 ARP, Reply _gateway is-at 52:55:0a:00:02:02 (oui Unknown), length 50
```

Et aussi avec la commande : **tcpdump -i enp0s3 icmp**
Cette commande a filtré les paquets ICMP (provenant des pings ou de traceroute) sur l'interface interne du routeur. Les captures ont montré que les paquets émis par le client étaient correctement acheminés via le routeur et transmis vers Internet.

```
root@vbox:/home/vboxuser# sudo tcpdump -i enp0s3 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:00:46.824953 IP vbox > dns.google: ICMP echo request, id 3529, seq 1, length 64
04:00:46.872942 IP dns.google > vbox: ICMP echo reply, id 3529, seq 1, length 64
04:00:47.827333 IP vbox > dns.google: ICMP echo request, id 3529, seq 2, length 64
04:00:47.870760 IP dns.google > vbox: ICMP echo reply, id 3529, seq 2, length 64
04:00:48.829164 IP vbox > dns.google: ICMP echo request, id 3529, seq 3, length 64
04:00:48.872719 IP dns.google > vbox: ICMP echo reply, id 3529, seq 3, length 64
04:00:49.830943 IP vbox > dns.google: ICMP echo request, id 3529, seq 4, length 64
04:00:49.876714 IP dns.google > vbox: ICMP echo reply, id 3529, seq 4, length 64
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
root@vbox:/home/vboxuser#
```

EXERCICE 3.13 :

L'analyse du trafic non crypté, tel que le protocole HTTP, à l'aide d'outils tels que tcpdump ou Wireshark sur un routeur, expose les informations sensibles échangées entre les utilisateurs et les serveurs. Cela ouvre la porte à des attaques de type "Man-in-the-Middle", où un attaquant peut intercepter, modifier et potentiellement compromettre l'intégrité des données. De plus, cette pratique peut violer la vie privée des utilisateurs et exposer des vulnérabilités de sécurité dans les applications. Pour renforcer la sécurité, il est vivement recommandé d'adopter des protocoles de communication cryptés tels que HTTPS, qui assurent la confidentialité et l'intégrité des données transitant sur le réseau.

EXERCICE 3.14 :

Pour tester la connexion FTP, un client FTP a été installé sur le client, et une connexion a été tentée vers un serveur FTP public, tel qu'un miroir de distribution Linux. La commande utilisée était la suivante :

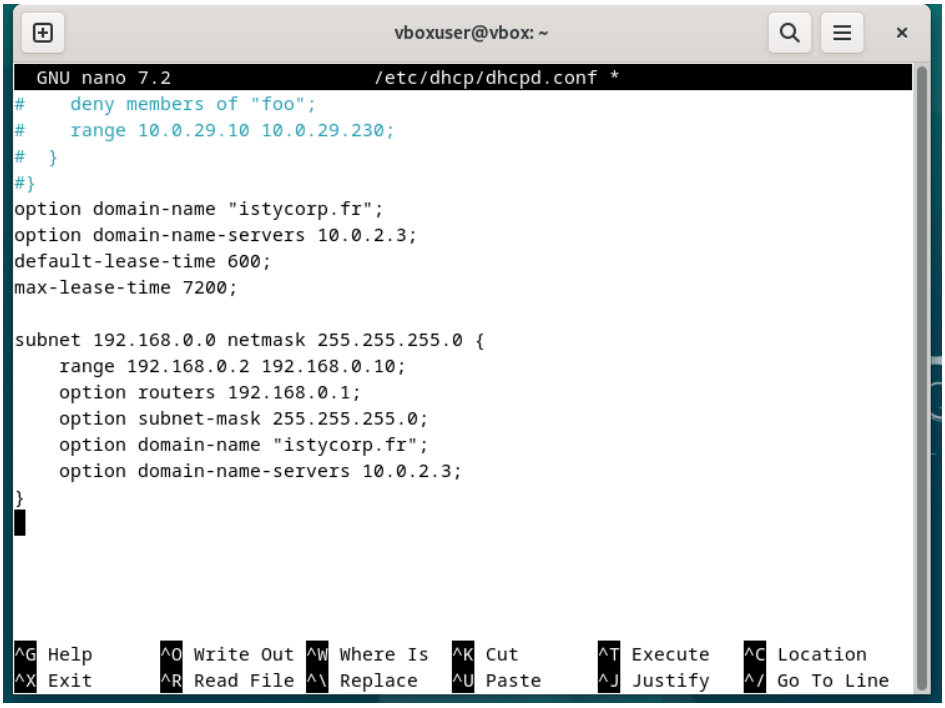
```
root@Client1:/home/vboxuser# ftp ftp.debian.org
Trying 146.75.54.132:21 ...
ftp: Can't connect to `146.75.54.132:21': Connection refused
Trying [2a04:4e42:7d::644]:21 ...
ftp: Can't connect to `2a04:4e42:7d::644:21': Network is unreachable
ftp: Can't connect to `ftp.debian.org:ftp'
ftp>
```

Je me suis donc intéressé à pourquoi est-ce que le mode actif de FTP ne fonctionnerai pas dans notre cas et j'en ai tiré les conclusions suivantes: Le mode actif de FTP est un des modes de transfert de données utilisés par le protocole FTP (File Transfer Protocol) pour transférer des fichiers entre un client FTP et un serveur FTP, il nécessite que le serveur se connecte directement au port ouvert sur le client, ce qui peut poser problème avec le NAT, qui modifie les adresses IP et les ports dans les paquets réseau.

Configuration dynamique : DHCP

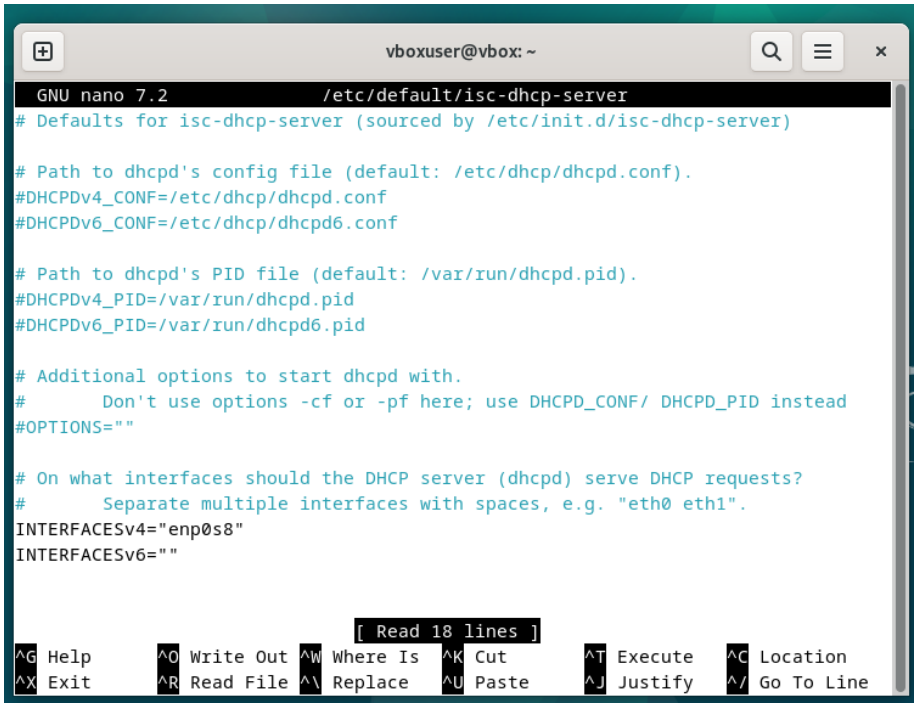
EXERCICE 3.15 :

Pour cet exercice, nous avons installé et configuré un serveur DHCP sur le routeur afin de permettre la configuration automatique des paramètres réseau pour les clients du réseau. Le serveur DHCP a été paramétré pour attribuer les adresses IP dans la plage 192.168.0.2 à 192.168.0.10. Le masque de sous-réseau utilisé est 255.255.255.0, ce qui correspond à une classe C standard, permettant de gérer jusqu'à 254 hôtes sur le réseau.



En plus de ces paramètres, la passerelle a été définie comme étant 192.168.0.1, ce qui correspond à l'adresse IP du routeur. Le domaine de recherche a été configuré comme "istycorp.fr", pour permettre une meilleure résolution des noms locaux.

Router (DHCP Running) :




```
root@vbox:/home/vboxuser# sudo systemctl restart isc-dhcp-server
root@vbox:/home/vboxuser# sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - LSB: DHCP server
     Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
     Active: active (running) since Wed 2024-12-25 04:26:57 +01; 10s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 4045 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
      Tasks: 1 (limit: 2284)
     Memory: 3.9M
        CPU: 36ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─4058 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf enp0s8

Dec 25 04:26:55 vbox systemd[1]: Starting isc-dhcp-server.service - LSB: DHCP server...
Dec 25 04:26:55 vbox isc-dhcp-server[4045]: Launching IPv4 server only.
Dec 25 04:26:55 vbox dhcpd[4058]: Wrote 0 leases to leases file.
Dec 25 04:26:55 vbox dhcpd[4058]: Server starting service.
Dec 25 04:26:57 vbox isc-dhcp-server[4045]: Starting ISC DHCPv4 server: dhcpd.
Dec 25 04:26:57 vbox systemd[1]: Started isc-dhcp-server.service - LSB: DHCP server.
```

Enfin, les serveurs DNS utilisés sont ceux spécifiés dans le fichier /etc/resolv.conf sur le routeur, ce qui garantit que les clients peuvent résoudre les noms de domaine externes correctement.

EXERCICE 3.16 :

Pour valider le fonctionnement du serveur DHCP configuré précédemment, nous avons redémarré la machine virtuelle client. Après le redémarrage, le client a automatiquement récupéré une adresse IP, un masque de sous-réseau, une passerelle par défaut et les paramètres DNS fournis par le serveur DHCP.

```
root@Client1:/home/vboxuser# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:55:55:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.3/24 brd 192.168.0.255 scope global dynamic enp0s3
       valid_lft 597sec preferred_lft 597sec
    inet6 fe80::a00:27ff:fe55:550f/64 scope link
       valid_lft forever preferred_lft forever
```

En vérifiant la configuration réseau avec la commande ip a, nous avons constaté que l'adresse IP attribuée au client se trouvait dans la plage définie (192.168.0.2 - 192.168.0.10). De plus, la passerelle par défaut et le serveur DNS étaient correctement configurés, comme spécifié dans les paramètres DHCP.

Le succès de cette validation montre que le serveur DHCP est opérationnel et distribue correctement les paramètres réseau aux clients. Cette automatisation facilite considérablement la gestion des configurations réseau, notamment dans un environnement avec de nombreux appareils.

Cache DNS :

EXERCICE 3.17 :

Pour installer les dnsmasq il faut exécuter la commande suivante :

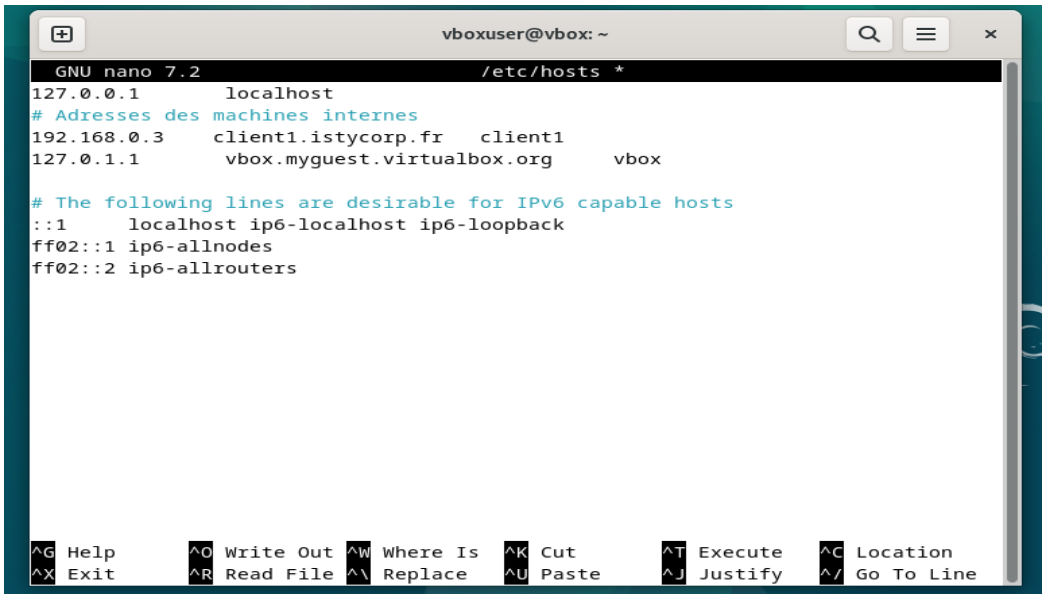
sudo apt install dnsmasq -y

Puis il suffit d'activer et de démarrer le service comme montrer dans le screen suivant :

```
root@vbox:/home/vboxuser# sudo systemctl restart dnsmasq
root@vbox:/home/vboxuser# sudo systemctl status dnsmasq
● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
     Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; preset: enab>
     Active: active (running) since Wed 2024-12-25 21:44:08 +01; 8s ago
    Process: 7921 ExecStartPre=/etc/init.d/dnsmasq checkconfig (code=exited, st>
    Process: 7930 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, stat>
    Process: 7939 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf (c>
   Main PID: 7938 (dnsmasq)
      Tasks: 1 (limit: 2284)
     Memory: 816.0K
        CPU: 46ms
     CGroup: /system.slice/dnsmasq.service
             └─7938 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7>

Dec 25 21:44:08 vbox systemd[1]: Starting dnsmasq.service - dnsmasq - A lightwe>
Dec 25 21:44:08 vbox dnsmasq[7938]: started, version 2.89 cachesize 1000
Dec 25 21:44:08 vbox dnsmasq[7938]: compile time options: IPv6 GNU-getopt DBus >
Dec 25 21:44:08 vbox dnsmasq[7938]: using nameserver 10.0.2.3#53
Dec 25 21:44:08 vbox dnsmasq[7938]: reading /etc/resolv.conf
Dec 25 21:44:08 vbox dnsmasq[7938]: using nameserver 10.0.2.3#53
Dec 25 21:44:08 vbox dnsmasq[7938]: using nameserver 10.0.2.3#53
Dec 25 21:44:08 vbox dnsmasq[7938]: read /etc/hosts - 9 names
Dec 25 21:44:08 vbox systemd[1]: Started dnsmasq.service - dnsmasq - A lightwei>
lines 1-22/22 (END)...skipping...
```

EXERCICE 3.18:

Pour définir proprement les résolutions de noms pour les machines du réseau internes, il faut éditer /etc/hosts sur le routeur et ajouter 192.168.0.3 client1 comme montré dans le screen suivant :



Pour tester il suffit de faire ping client 1 :

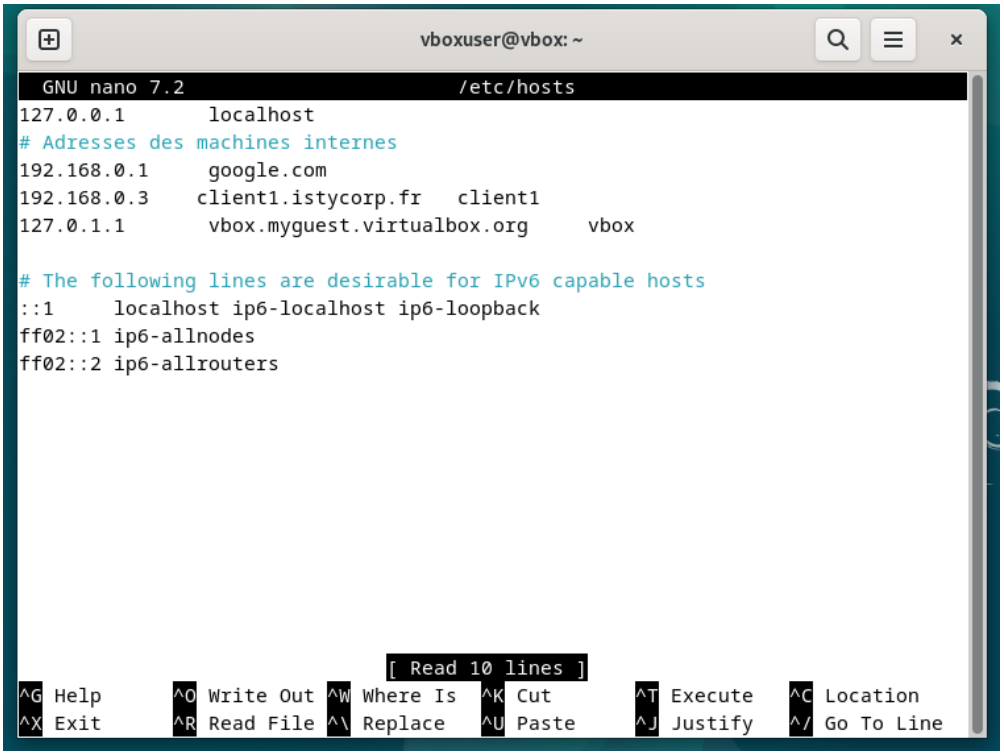
```
root@vbox:/etc/systemd/network# ping client1
PING client1.istycorp.fr (192.168.0.3) 56(84) bytes of data.
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=1 ttl=64 time=0.485 ms
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=2 ttl=64 time=0.576 ms
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=3 ttl=64 time=0.539 ms
64 bytes from client1.istycorp.fr (192.168.0.3): icmp_seq=4 ttl=64 time=0.526 ms
^C
--- client1.istycorp.fr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.485/0.531/0.576/0.032 ms
```

Pour vérifier la propagation il suffit d'exécuter: ping client1 sur le client et en effet cela marche

```
root@Client1:/etc/systemd/network# ping client1
PING Client1.local (127.0.1.1) 56(84) bytes of data.
64 bytes from Client1.local (127.0.1.1): icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from Client1.local (127.0.1.1): icmp_seq=2 ttl=64 time=0.031 ms
64 bytes from Client1.local (127.0.1.1): icmp_seq=3 ttl=64 time=0.032 ms
64 bytes from Client1.local (127.0.1.1): icmp_seq=4 ttl=64 time=0.029 ms
^C
--- Client1.local ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 0.029/0.031/0.033/0.001 ms
```

EXERCICE 3.19:

Pour cela, j'édite a nouveau **/etc/hosts** et je redirige la résolution de google.com vers mon routeur :



Et je teste sur le **server :**

```
root@vbox:/etc/systemd/network# ping google.com
PING google.com (192.168.0.1) 56(84) bytes of data.
64 bytes from google.com (192.168.0.1): icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from google.com (192.168.0.1): icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from google.com (192.168.0.1): icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from google.com (192.168.0.1): icmp_seq=4 ttl=64 time=0.057 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.039/0.048/0.057/0.006 ms
```

Et sur le client :

```
root@Client1:/etc/network# ping -4 google.com
PING google.com (192.168.0.1) 56(84) bytes of data.
64 bytes from _gateway (192.168.0.1): icmp_seq=1 ttl=64 time=0.410 ms
64 bytes from _gateway (192.168.0.1): icmp_seq=2 ttl=64 time=0.524 ms
64 bytes from _gateway (192.168.0.1): icmp_seq=3 ttl=64 time=0.481 ms
64 bytes from _gateway (192.168.0.1): icmp_seq=4 ttl=64 time=0.540 ms
64 bytes from _gateway (192.168.0.1): icmp_seq=5 ttl=64 time=0.529 ms
64 bytes from _gateway (192.168.0.1): icmp_seq=6 ttl=64 time=0.424 ms
64 bytes from _gateway (192.168.0.1): icmp_seq=7 ttl=64 time=0.435 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6139ms
rtt min/avg/max/mdev = 0.410/0.477/0.540/0.050 ms
root@Client1:/etc/network# nslookup google.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Name:   google.com
Address: 192.168.0.1
Name:   google.com
Address: 2a00:1450:4003:80c::200e
```

Les résultats montrent que nslookup résout correctement google.com en l'adresse 192.168.0.1 (redirection définie dans /etc/hosts sur le routeur),

La sécurité des traductions d'adresse sur un réseau dont on n'est pas maître présente des vulnérabilités importantes. Les risques incluent des attaques de détournement DNS permettant la redirection vers des sites malveillants, des attaques Man-in-the-Middle exploitant les mécanismes de translation d'adresses, des problèmes de filtrage de sécurité, et des altérations potentielles du trafic.

Pour renforcer la sécurité, :

- L'utilisation de protocoles DNS sécurisés tels que DoH (DNS over HTTPS) ou DoT (DNS over TLS) contribue à chiffrer les requêtes DNS, réduisant ainsi le risque de manipulation.
- La mise en place de technologies VPN permet de chiffrer l'ensemble du trafic entre l'utilisateur et le serveur, renforçant la confidentialité et la sécurité des données transitant sur le réseau,
- La surveillance continue du réseau

- L'éducation des utilisateurs sur les bonnes pratiques de sécurité sont essentielles.
- La collaboration avec les administrateurs réseau est également cruciale pour mettre en œuvre des politiques de sécurité robustes et atténuer les risques associés à la traduction d'adresses dans des environnements réseau non maîtres.

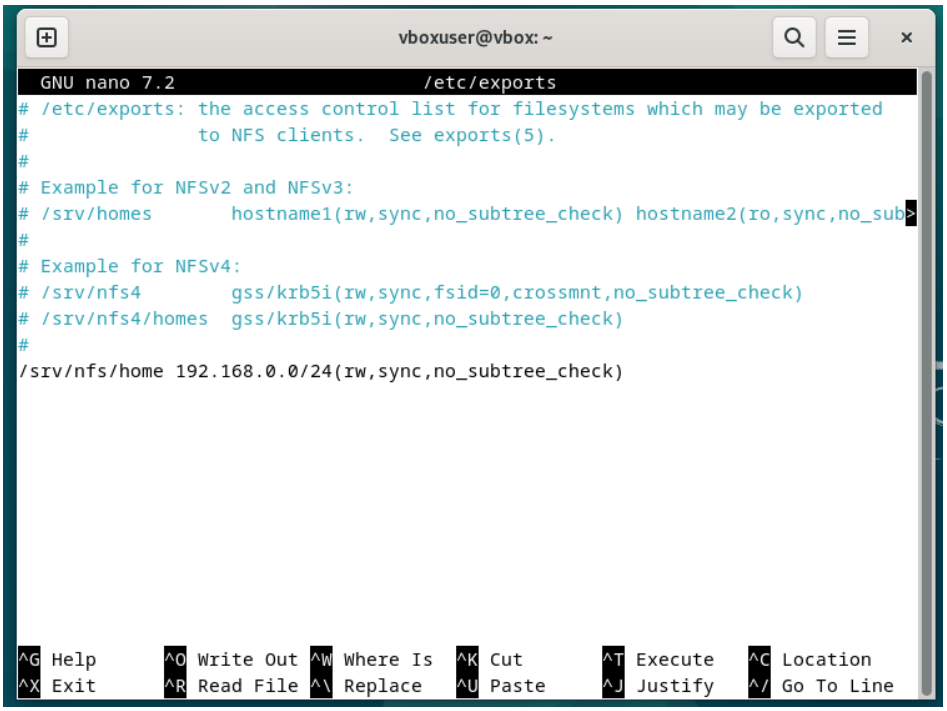
Bonus : serveur NFS

EXERCICE 3.20:

Pour répondre à cette question, nous avons mis en place un serveur **NFS (Network File System)** sur le routeur afin de centraliser le dossier **home** et le rendre accessible au client. Cette configuration permet de partager des fichiers entre plusieurs machines dans un environnement réseau, en facilitant la collaboration et en réduisant la redondance des données comme montré dans le screen suivant :

```
root@vbox:/home/vboxuser# sudo mkdir -p /srv/nfs/home
root@vbox:/home/vboxuser# sudo chown nobody:nogroup /srv/nfs/home
sudo chmod 755 /srv/nfs/home
root@vbox:/home/vboxuser# sudo nano /etc/exports
root@vbox:/home/vboxuser# sudo systemctl restart nfs-kernel-server
root@vbox:/home/vboxuser# sudo exportfs -v
/srv/nfs/home   192.168.0.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
```

Nous avons tout d'abord installé le serveur **NFS** sur le routeur et configuré le fichier /etc/exports pour spécifier les dossiers à partager. Par exemple, nous avons exporté le répertoire /srv/nfs/home en le rendant accessible pour les machines du réseau local, avec les permissions adéquates. Ensuite, nous avons redémarré les services NFS pour prendre en compte cette configuration.



On redémarre le service avec la commande **systemctl restart nfs-kernel-server** et puis on vérifie que la partage est actif

```
root@vbox:/home/vboxuser# sudo systemctl restart nfs-kernel-server
root@vbox:/home/vboxuser# sudo exportfs -v
/srv/nfs/home   192.168.0.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
```

Sur le client, nous avons créé un point de montage local, généralement /mnt/nfs_home, pour accéder au dossier partagé. Nous avons utilisé la commande mount pour monter le partage NFS manuellement, et nous avons rendu le montage persistant en ajoutant une entrée correspondante dans le fichier /etc/fstab. Une fois le partage monté, nous avons vérifié que le client peut accéder au contenu du dossier partagé sur le routeur.

```
root@vbox:/home/vboxuser# sudo mkdir -p /mnt/nfs_home
root@vbox:/home/vboxuser# sudo mount 192.168.0.1:/srv/nfs/home /mnt/nfs_home
root@vbox:/home/vboxuser# df -h
Filesystem                 Size  Used Avail Use% Mounted on
udev                       953M     0  953M   0% /dev
tmpfs                      197M  1.3M  196M   1% /run
/dev/sda1                  8.9G  5.5G  3.0G  66% /
tmpfs                      984M     0  984M   0% /dev/shm
tmpfs                      5.0M  8.0K  5.0M   1% /run/lock
tmpfs                      197M   96K  197M   1% /run/user/1000
192.168.0.1:/srv/nfs/home  8.9G  5.5G  3.0G  66% /mnt/nfs_home
```

On vérifie que le partage est monté :

```
mount: (hint) your fstab has been modified, but systemd still uses
       the old version; use 'systemctl daemon-reload' to reload.
root@Client1:/etc/network# systemctl daemon-reload
root@Client1:/etc/network# df -h
Filesystem                 Size  Used Avail Use% Mounted on
udev                       953M     0  953M   0% /dev
tmpfs                      197M  1.2M  196M   1% /run
/dev/sda1                  8.9G  5.4G  3.0G  65% /
tmpfs                      984M     0  984M   0% /dev/shm
tmpfs                      5.0M  8.0K  5.0M   1% /run/lock
tmpfs                      197M   96K  197M   1% /run/user/1000
192.168.0.1:/srv/nfs/home  8.9G  5.5G  3.0G  66% /mnt/nfs_home
root@Client1:/etc/network#
```

Et enfin on teste l'accès avec une création d'un fichier sur le répertoire monté

```
root@Client1:/mnt/nfs_home# ls
test_file
```

EXERCICE 3.21:

Dans une configuration NFS, la sécurité des droits utilisateurs est cruciale pour maintenir un environnement sécurisé. Il est impératif de synchroniser les UID et GID entre le serveur et le client, de configurer correctement les droits d'accès aux fichiers partagés, et d'utiliser des options telles que pare-feu pour restreindre l'accès aux

adresses IP autorisées. La gestion appropriée des utilisateurs, la journalisation des activités NFS, et l'application de bonnes pratiques telles que la limitation des privilèges root contribuent à renforcer la sécurité. Une attention particulière à la configuration du fichier `/etc/exports` et l'exploration d'options comme le chiffrement renforcent la robustesse de la sécurité NFS. La vigilance continue, la surveillance des journaux système et l'application des correctifs de sécurité sont essentielles pour garantir un partage de fichiers fiable et sécurisé.

Bonus : redirection SSH

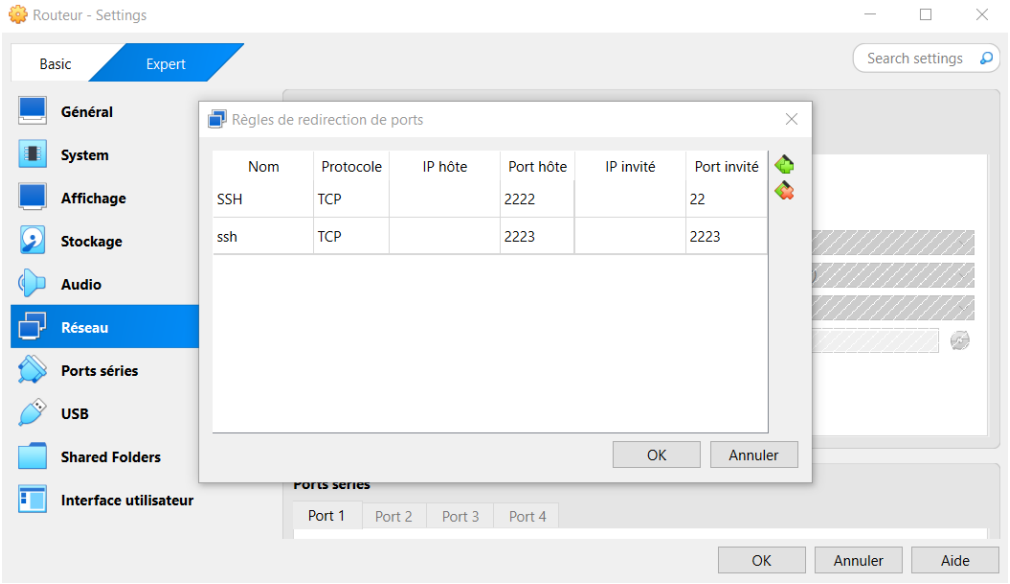
EXERCICE 3.22 :

Pour répondre au bonus sur la redirection **SSH**, nous avons configuré un serveur **SSH** sur le client pour permettre une connexion à distance. Cette opération nécessite une redirection du port au niveau du NAT sur le routeur VirtualBox et du routeur lui-même.

D'abord Nous avons installé le serveur **OpenSSH** sur le client avec la commande sudo apt install openssh-server. Après l'installation, nous avons vérifié que le service **SSH** est actif en utilisant sudo systemctl status ssh.

```
root@Client1:/etc/network# sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
root@Client1:/etc/network# sudo systemctl start ssh
root@Client1:/etc/network# sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-12-25 23:49:25 +01; 1min 3s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 7671 (sshd)
      Tasks: 1 (limit: 2284)
     Memory: 1.4M
        CPU: 29ms
     CGroup: /system.slice/ssh.service
             └─7671 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 25 23:49:25 Client1 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Dec 25 23:49:25 Client1 sshd[7671]: Server listening on 0.0.0.0 port 22.
Dec 25 23:49:25 Client1 sshd[7671]: Server listening on :: port 22.
Dec 25 23:49:25 Client1 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@Client1:/etc/network#
```

Ensuite, Nous avons configuré une règle de redirection de port sur l'interface NAT du routeur dans VirtualBox. La règle redirige les connexions entrantes vers le port 2223 de l'hôte vers le port 22 du client via le routeur. Cela permet à une machine externe d'accéder au client en passant par le routeur.



Puis on ajout d'une règle iptables pour le NAT sur le routeur avec une règle iptables a sur le routeur pour rediriger les paquets du port 2223 vers le port 22 du client avec la commande suivante :

```
root@vbox:/home/vboxuser# sudo iptables -t nat -A PREROUTING -p tcp --dport 2223 -j DNAT --to-destination 192.168.0.3:22
root@vbox:/home/vboxuser# sudo iptables -A FORWARD -p tcp -d 192.168.0.3 --dport 22 -j ACCEPT
root@vbox:/home/vboxuser# sudo netfilter-persistent save
sudo netfilter-persistent reload
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
root@vbox:/home/vboxuser#
```

Pour valider, nous avons utilisé la commande SSH depuis l'hôte pour établir une connexion avec le client via le port 2223 :

```
C:\Users\LENOVO>ssh -p 2223 vboxuser@127.0.0.1
The authenticity of host '[127.0.0.1]:2223 ([127.0.0.1]:2223)' can't be established.
ECDSA key fingerprint is SHA256:5+gaautTrLQ1LoSSOP02SBNTBkapqVLSQw0cuC62WtM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[127.0.0.1]:2223' (ECDSA) to the list of known hosts.
vboxuser@127.0.0.1's password:
Linux Client1 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) 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.
vboxuser@Client1:~$
```

La connexion a été établie avec succès, prouvant que la redirection est fonctionnelle.