Réseau

- Serveur DHCP (ISC-DHCP-SERVER)
- Serveur PXE (TFTP & ISC-DHCP)
- WireGuard Client Installation
- Ajout de routes statiques
- Tunneling SSH
- Ping
- Configuration /etc/network/interfaces
- Iptables

Serveur DHCP (ISC-DHCP-SERVER)

Configuration d'un serveur DHCP avec routage et NAT

Mise à jour du système

Commencez par mettre à jour votre système : **`sudo apt update && sudo apt upgrade`**

Installation du service DHCP

Installez le service DHCP avec la commande suivante : **`sudo apt install isc-dhcp-server`**

Configuration de l'interface réseau

Attribuez une adresse IP statique à l'interface réseau que vous souhaitez utiliser pour le DHCP. Par exemple, pour Debian, éditez le fichier `/etc/network/interfaces`. Si votre gestionnaire de réseau est différent, adaptez cette étape selon vos besoins.

```
auto eth1
iface eth1 inet static
    address 192.168.2.1
    netmask 255.255.255.0
    network 192.168.2.0
    broadcast 192.168.2.255
```

"***eth1***" étant l'interface qui diffusera le DHCP.

Configuration du fichier DHCP

Ouvrez le fichier de configuration principal du service DHCP : **`vim /etc/dhcp/dhcpd.conf`** Recherchez les lignes suivantes et modifiez-les selon vos besoins ou laissez-les par défaut.Ajoutez ensuite la configuration de votre sous-réseau.

Voici un exemple :

```
subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.2 192.168.1.200;
   option routers 192.168.1.1;
   option subnet-mask 255.255.255.0;
   option domain-name-servers 8.8.8.8, 8.8.4.4;
   option domain-name "local";
}
```

Modifiez cette configuration en fonction de vos besoins spécifiques.

Spécification de l'interface réseau pour le DHCP

Éditez le fichier suivant pour indiquer l'interface réseau que le service DHCP doit écouter : **`vim /etc/default/isc-dhcp-server`**

Ajoutez ou modifiez la ligne suivante pour spécifier votre interface réseau (l'interface qui diffusera le DHCP) :

```
INTERFACESv4="eth1"
```

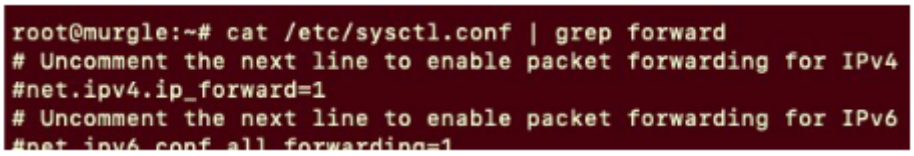
Redémarrage et activation du service DHCP

Redémarrez le service DHCP et configurez-le pour qu'il démarre automatiquement au démarrage du système : **`systemctl restart isc-dhcp-server`**
`systemctl enable isc-dhcp-server`

Activation du routage IP

Pour activer le routage entre les interfaces réseau, éditez le fichier suivant : **`vim /etc/sysctl.conf`**
Ajoutez ou décommentez la ligne suivante :

net.ipv4.ip_forward = 1



Appliquez les modifications avec la commande suivante : **`sudo sysctl -p`**

Configuration du NAT avec iptables

Configurez le NAT (Network Address Translation) pour permettre aux machines connectées via DHCP d'accéder à Internet via l'interface réseau de votre serveur.Ajoutez cette règle iptables (remplacez `eth0` par l'interface réseau utilisée pour l'accès Internet) : **`sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`** **(à la place de eth0 mettez l'interface qui fournit internet à votre serveur)** Pour rendre cette règle permanente, installez **iptables-persistent** : **`sudo apt install iptables-persistent`** Lors de l'installation, acceptez de sauvegarder les règles actuelles pour IPv4 et IPv6 (si applicable). Ensuite, sauvegardez manuellement les règles avec la commande suivante : **`sudo iptables-save > /etc/iptables/rules.v4`** Vérifiez que les modifications ont bien été enregistrées : **`cat /etc/iptables/rules.v4`**

Activez et redémarrez le service netfilter-persistent pour appliquer les règles au démarrage : **`sudo systemctl enable netfilter-persistent`**
`sudo systemctl restart netfilter-persistent` **Votre serveur DHCP avec routage et NAT est**

maintenant configuré et prêt à être utilisé !

Serveur PXE (TFTP & ISC-DHCP)

Avant de suivre ce tutoriel, vous devez disposer d'un **serveur ISC-DHCP fonctionnel** sur le même serveur où vous souhaitez installer le PXE. Assurez-vous également d'avoir effectué une mise à jour du système avec la commande suivante : **sudo apt update && sudo apt upgrade**

Installation des utilitaires nécessaires

Installez les utilitaires Linux nécessaires ainsi qu'un serveur TFTP avec la commande suivante : **sudo apt install syslinux-common syslinux-efi tftpd-hpa**

Création du répertoire de configuration

Créez un répertoire pour stocker les fichiers de configuration :** sudo mkdir /tftpboot **Accédez au répertoire /usr/lib/syslinux/modules/efi64/ et copiez les fichiers suivants dans /tftpboot :** cp {ldlinux.e64,libutil.c32,menu.c32} /tftpboot **Copiez également le fichier suivant :** cp /usr/lib/SYSLINUX.EFI/efi64/syslinux.efi /tftpboot **Accédez au répertoire /tftpboot et créez un sous-répertoire pxelinux.cfg : **cd /tftpboot**
mkdir pxelinux.cfg

Configuration du serveur TFTP

Modifiez le fichier de configuration TFTP avec la commande suivante :** vim /etc/default/tftpd-hpa **
Assurez-vous que le contenu du fichier est conforme à ce qui suit :

```
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
```

N'oubliez pas de configurer votre pare-feu pour autoriser le port **69/UDP** si nécessaire. Démarrez et activez le service TFTP : `systemctl start tftpd-hpa`
`systemctl enable tftpd-hpa`

Configuration du serveur DHCP

Ouvrez le fichier de configuration DHCP situé à l'adresse suivante : `/etc/dhcp/dhcpd.conf`. Ajoutez ces deux lignes à la configuration de votre sous-réseau DHCP :

```
next-server 192.168.1.5;
option bootfile-name "syslinux.efi";
```

Remplacez `192.168.1.5` par l'adresse IP de votre serveur PXE.

Exemple de configuration DHCP complète :

```
subnet 192.168.0.0 netmask 255.255.255.0 {
   range 192.168.0.106 192.168.0.200;
   option routers 192.168.0.1;
   default-lease-time 3600;
   max-lease-time 86400;
   next-server 192.168.0.105;
   option bootfile-name "syslinux.efi";
}
```

Redémarrez le service DHCP pour appliquer les modifications : `systemctl restart isc-dhcp-server`

Ajout des fichiers d'installation Debian

Créez un répertoire pour l'ISO Debian dans `/tftpboot` : `sudo mkdir /tftpboot/Debian` Montez votre fichier ISO Debian et copiez son contenu dans le répertoire créé précédemment :

`sudo mount your_iso_file.iso /mnt`
`sudo cp -r /mnt/* /tftpboot/Debian`

> Cette commande peut prendre du temps en fonction de la taille de votre fichier ISO.

Configuration du menu PXE

Accédez au répertoire `/tftpboot/pxelinux.cfg` et créez un fichier nommé `default` : `vim /tftpboot/pxelinux.cfg/default` Ajoutez la configuration suivante pour Debian Net Boot :

UI menu.c32

LABEL Debian Net Boot
MENU LABEL Debian
KERNEL debian/install.amd/vmlinuz
APPEND initrd=Debian/install.amd/initrd.gz
TEXT HELP
Installation minimale de Debian.
ENDTEXT

Les fichiers spécifiés (vmlinuz et initrd.gz) doivent être situés dans le répertoire `/tftpboot/Debian/install.amd`.

Exemple d'arborescence des fichiers dans /tftpboot (peuvent être différent selon les images) :

/tftpboot/
├── Debian/
│ ├── install.amd/
│ │ ├── vmlinuz
│ │ ├── initrd.gz
├── ldlinux.e64
├── libutil.c32
├── menu.c32
├── pxelinux.cfg/
│ ├── default
├── syslinux.efi

Ajout d'autres systèmes d'exploitation (exemple Ubuntu)

Pour ajouter un autre système, comme Ubuntu, utilisez une configuration similaire en modifiant les chemins et noms des fichiers correspondants.Exemple pour Ubuntu :

UI menu.c32

LABEL Ubuntu Net Boot
MENU LABEL Ubuntu
KERNEL ubuntu/install/amd64/linux
APPEND initrd=ubuntu/install/amd64/initrd.gz
TEXT HELP
Installation minimale d'Ubuntu.
ENDTEXT

Finalisation

Redémarrez votre service TFTP pour finaliser la configuration : **`systemctl restart tftpd-hpa`**

Votre serveur PXE est maintenant configuré et prêt à être utilisé !

WireGuard Client Installation

ÉTAPE 1 : INSTALLER WIREGUARD

1. Mettez à jour votre système :

   `sudo apt update && sudo apt upgrade -y`
2. Installez WireGuard :

   `sudo apt install wireguard`

> **Si vous n'avez pas le paquet resolvconf d'installer installez le également avec : " apt install resolvconf -y "**

ÉTAPE 2 : GÉNÉRER LES CLÉS POUR LE CLIENT

WireGuard utilise une paire de clés publique/privée pour l'authentification.

1. Accédez au répertoire de configuration de WireGuard :

   `cd /etc/wireguard`
2. Générez les clés :

   ```
   umask 077
   wg genkey | tee private.key | wg pubkey > public.key
   ```
3. Vérifiez les clés générées :
   - Clé privée :

     `cat private.key`
   - Clé publique :

`cat public.key`

4. Conservez la clé publique, car elle devra être partagée avec le serveur WireGuard.

ÉTAPE 3 : CRÉER LE FICHIER DE CONFIGURATION DU CLIENT

1. Créez un fichier de configuration pour l'interface WireGuard (par exemple `wg0.conf`) :

   `sudo nano /etc/wireguard/wg0.conf`
2. Ajoutez le contenu suivant au fichier (adaptez selon vos besoins) :

   ```
   [Interface]
   PrivateKey = <clé_privée_du_client> # Remplacez par la clé privée générée (fichier private.key)
   Address = 192.168.110.2/24          # Adresse IP privée du client dans le tunnel VPN
   DNS = 192.168.110.1                 # (Optionnel) Serveur DNS à utiliser via le VPN

   [Peer]
   PublicKey = <clé_publique_du_serveur> # Clé publique fournie par le serveur WireGuard
   Endpoint = <ip_publique_du_serveur>:51820 # Adresse IP ou nom de domaine et port du serveur WireGuard
   AllowedIPs = 0.0.0.0/0, ::/0          # Acheminer tout le trafic via le VPN
   ```
3. Remplacez `<clé_privée_du_client>` et `<clé_publique_du_serveur>` par les valeurs appropriées.
4. Sauvegardez et fermez le fichier (`Ctrl+O`, puis `Ctrl+X`).

ÉTAPE 4 : ACTIVER ET TESTER LA CONNEXION

1. Activez l'interface WireGuard :

   `sudo wg-quick up wg0`
2. Vérifiez que l'interface est active :

   `ip a show wg0`
3. Testez la connectivité en pingant une adresse IP via le VPN (par exemple, l'adresse IP privée du serveur ou une ressource accessible uniquement via le VPN) :

   `ping 192.168.110.1`

1. Pour désactiver la connexion VPN :

   `sudo wg-quick down wg0`

ÉTAPE 5 : ACTIVER LE DÉMARRAGE AUTOMATIQUE (OPTIONNEL)

Si vous souhaitez que l'interface VPN démarre automatiquement au démarrage du système :

1. Activez le service WireGuard associé à `wg0` :

   `sudo systemctl enable wg-quick@wg0.service`
2. Désactivez-le si nécessaire :

   `sudo systemctl disable wg-quick@wg0.service`

NOTES IMPORTANTES

- **Clé publique du client** : Vous devrez fournir cette clé au serveur WireGuard pour qu'il puisse autoriser votre connexion.
  - Pour afficher la clé publique du client, utilisez :

    `cat /etc/wireguard/public.key`
- **Sécurité des clés** : Assurez-vous que les fichiers contenant vos clés privées sont protégés avec des permissions strictes.

  `chmod 600 /etc/wireguard/private.key /etc/wireguard/wg0.conf`

Dans le fichier /etc/wireguard/wg0.conf, vous pouvez rajouter ceci à la fin si vous voulez maintenir une connexion permanente au vpn PersistentKeepalive = 25

Cela envoie un paquet keepalive toutes les 25 secondes pour maintenir la connexion active.
Avec ces étapes, votre machine Linux sera configurée comme un client VPN WireGuard et pourra se connecter à un serveur WireGuard distant pour sécuriser ses communications réseau !

Ajout de routes statiques

Exemple de configuration Pour une config Netplan

Configuration des routes réseau

Netplan (`/etc/netplan/`)

Les routes doivent être définies directement sous les paramètres de l'interface de sortie souhaitée. Il est important de choisir le bon fichier netplan avant d'y placer vos routes.

Configuration avec IP statique

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp114s0:
      addresses:
        - 192.168.249.8/24
      gateway4: 192.168.249.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

```
    dhcp4: no
    dhcp6: no
```

Configuration avec route statique (IP statique)

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:
      addresses:
        - 192.168.1.100/24
      routes:
        - to: 192.168.1.82/32
          via: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
```

Configuration avec route statique (DHCP)

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ens33:
      dhcp4: yes
      routes:
        - to: 192.168.1.82/32
          via: 192.168.1.1
```

Explication des attributs

- **to** : l'adresse IP de destination à atteindre (ex: `192.168.1.82/32` pour une IP précise, ou `192.168.45.0/24` pour un réseau entier).
- **via** : la gateway permettant d'atteindre la destination. Dans le cas d'un réseau sans gateway, on indiquera l'adresse IP de l'interface de sortie.

Cas particulier : route sans gateway (`on-link: true`)

Lorsque vous avez **deux interfaces sur le même subnet**, le kernel peut ne pas savoir quelle interface utiliser pour joindre une IP spécifique. Dans ce cas, `on-link: true` combiné à une table de routage dédiée permet de forcer l'interface de sortie.

`on-link: true` indique au kernel que la destination est directement accessible sur le lien réseau, sans gateway intermédiaire. La valeur de `via` sera alors l'IP de destination elle-même.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enxdceae7dd1709:
      addresses:
        - 192.168.1.244/24
      routes:
        - to: 192.168.1.15/32
          via: 192.168.1.15
          on-link: true
          table: 100
        - to: 192.168.1.22/32
          via: 192.168.1.22
          on-link: true
          table: 100
      routing-policy:
        - to: 192.168.1.15/32
          table: 100
          priority: 50
        - to: 192.168.1.22/32
          table: 100
          priority: 50
      dhcp4: false
```

> **Note :** Plus la valeur de `priority` est basse, plus la règle est prioritaire. Une priorité de `50` sera donc appliquée avant une règle à `100`.

> **Attention :** Cette configuration ne résout pas le cas où l'IP de destination est aussi assignée localement sur une autre interface de la même machine. Dans ce cas, le kernel traitera toujours le paquet en local.

/etc/network/interfaces

Ajouter la commande suivante après la configuration réseau dans le fichier `/etc/network/interfaces` :

```
up ip route add <réseau_destination>/<masque> via <gateway_ou_interface>
```

Exemple :

```
up ip route add 192.168.45.0/24 via 192.168.6.1
```

Tunneling SSH

Le protocol SSH nous permet de créer des tunnels pour accéder à des appareil distant, à condition d'avoir une connexion SSH sur un appareil au sein du réseau.

SSH Local Port Forwarding (-L)

- Concept : Relie un port spécifique de votre machine à un port spécifique d'une machine distante via le serveur SSH.
- Commande :

```
ssh -L [PORT_LOCAL]:[IP_DESTINATION]:[PORT_DESTINATION] utilisateur@serveur_ssh
```

- Exemple : `ssh -L 8080:192.168.1.1:80 user@remote`
  - **Accès** : Tapez http://localhost:8080 pour voir l'interface du routeur 192.168.1.1

SSH Dynamic Port Forwarding (-D)

- Concept : Transforme votre connexion SSH en un Proxy SOCKS. Au lieu de viser une seule IP, votre navigateur peut atteindre n'importe quelle IP du réseau distant.
- Commande:

```
ssh -D [PORT_PROXY] utilisateur@serveur_ssh
```

- Exemple : `ssh -D 1080 user@remote` Vous pouvez aussi le lancer en arrière plan avec la commande `ssh -f -N -D 1080 utilisateur@serveur_ssh`
  - `-f` : Met SSH en arrière plan
  - `-N` : N'ouvre pas de termanl distant (uniquement le tunnel)

Sur macOS, une fois le tunnel ouvert vous pouvez ouvrir un navigateur avec :

```
open -a "Google Chrome" --args --proxy-server="socks5://localhost:1080"
```

- Ça fonctionne avec tous les navigateurs basé sur chronium comme Arc par exemple.
  - Exemple avec Arc `open -a "Arc" --args --proxy-server="socks5://localhost:1080"`

Une fois le nagivateur ouvert, tapez juste l'ip voulue pour y accéder. Si vous voulez changer le port ne le changer pas sur la commande faites juste `adresse_ip:port`.

Le port 1080 sert uniquement de porte d'entrée.

- Pour fermer le tunnel vous pouvez utiliser la commande `pkill -f "ssh -D 1080"` qui stoppera tous les commandes contenant `ssh -D 1080`

Ping

La commande ping utilise le protocole ICMP pour vérifier la connectivité avec une machine distante.

□□ Récapitulatif des options `ping` essentielles

Option	Nom officiel	Description & Usage	Exemple
`-c`	**Count**	Arrête le ping après un nombre précis de paquets. Indispensable pour les scripts.	`ping -c 4 8.8.8.8`
`-I`	**Interface**	Force le départ du paquet via une interface (eth0) ou une adresse IP source.	`ping -I wlan0 1.1.1.1`
`-i`	**Interval**	Définit le délai entre chaque envoi (en secondes). Par défaut : 1s.	`ping -i 0.2 10.0.0.1`
`-s`	**Size**	Définit la taille du paquet en octets (utile pour tester le MTU).	`ping -s 1472 8.8.8.8`
`-W`	**Timeout**	Temps d'attente max pour une réponse (en secondes) avant de considérer l'échec.	`ping -W 2 192.168.1.1`
`-f`	**Flood**	Envoie les paquets au maximum de la capacité réseau (nécessite `sudo`).	`sudo ping -f 1.1.1.1`
`-a`	**Audible**	Émet un bip sonore (PC speaker) à chaque réponse reçue.	`ping -a 192.168.1.50`
`-n`	**Numeric**	Désactive la résolution DNS pour un démarrage instantané.	`ping -n google.com`

Configuration /etc/network/interfaces

Sur debian par défaut les paramètres de configuration réseau se trouve dans /etc/network/interfaces.

Voici un exemple de configuration statique.

Activation au branchement/détection

allow-hotplug ens18

Configuration manuelle (static)

```
iface ens18 inet static
    address 192.168.1.50
    netmask 255.255.255.0
    gateway 192.168.1.1
    # Optionnel : serveurs DNS (Google et Cloudflare ici)
    dns-nameservers 8.8.8.8 1.1.1.1
```

Iptables

Iptables est un utilitaire permettant de maniper le firewall natif de linux (netfilter).

Ce guide récapitule les commandes essentielles pour administrer un pare-feu Linux.

Pour changer les policies par défaut on peut utiliser les commandes:

```
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT
```

Tout trafic ne s'appliquant pas à une règle présente suivra par défaut la policy.

1. Consultation des règles

Commande	Description
`iptables -L`	Liste les règles de la table **filter** (par défaut).
`iptables -L -n -v`	Liste détaillée (octets, paquets) sans résolution DNS.
`iptables -L --line-numbers`	Affiche les numéros de ligne (utile pour supprimer).
`iptables -S`	Affiche les règles sous forme de commandes brutes.
`iptables -t nat -L -n -v`	Liste les règles de la table **NAT** (redirections).

---

2. Gestion des règles (Ajout/Suppression)

> ⚠ **Note :** L'ordre des règles est crucial, la lecture se fait de haut en bas.

- **Ajouter à la fin (`-A`)** :
  `iptables -A INPUT -p tcp --dport 80 -j ACCEPT`
- **Insérer au début ou à une ligne précise (`-I`)** :
  `iptables -I INPUT 1 -i lo -j ACCEPT`
- **Supprimer par numéro de ligne (`-D`)** :
  `iptables -D INPUT 3`

- **Vider toutes les règles d'une table (`-F`)** :
  `iptables -F`

---

3. Configuration du NAT (Table NAT)

Le NAT s'applique toujours avec l'option `-t nat`.

Partage de connexion (Masquerade / SNAT)

Permet aux hôtes internes (LAN) d'accéder à Internet via l'IP du pare-feu.

```
iptables -t nat -A POSTROUTING -o [interface_WAN] -j MASQUERADE
```