

## **Administration système sous Linux**

**L1 TR/MIC EC2LT 2020 – 2021**

**MAJ : 24.06.2021**

### **Activité 4 : Sécurité des services et du réseau**

#### **Objectifs :**

A la fin de cette activité, vous devez être capable de :

- Comprendre les objectifs principaux de la sécurité des systèmes d'information
- Décrire les menaces principales qui pèsent sur un réseau
- Expliquer les principales techniques de protection d'un réseau contre les attaques
- Connaître les ports associés aux principaux services d'un réseau
- Utiliser l'outil nmap pour détecter les ports et les services ouverts sur votre machine et sur les autres machines du réseau
- Comprendre le principe de fonctionnement des pare-feu et mettre en place des restrictions d'accès avec les fichiers /etc/hosts.deny et /etc/hosts.allow de Linux
- Utiliser netfilter et son interface iptables pour filtrer les paquets à destination et quittant votre réseau
- De transformer votre machine Linux en routeur et mettre en place le mécanisme de translation d'adresses (NAT) avec iptables
- Comprendre et savoir mettre en œuvre les différents types de NAT (statique, dynamique, PAT)
- Configurer un point d'accès à l'aide de son interface graphique d'administration

## **INTRODUCTION A LA SECURITE DES SYSTEMES D'INFORMATION**

Le terme sécurité évoque toute une série de concepts allant de la protection des données à la prévention contre les accès frauduleux.

En termes de sécurité réseau, le plus important pour une organisation est de déterminer ce qui est considéré comme crucial pour le bon fonctionnement de son système d'information et de s'assurer que ces éléments en sécurité.

En d'autres termes, il faut définir une politique de sécurité adaptée à son organisation et à ses besoins. Les entreprises ont généralement des données privées et d'autres qui sont destinées au public. Il est crucial que la distinction soit clairement établie entre ces deux catégories, et que l'implémentation du réseau reflète cette séparation en restreignant l'accès aux données sensibles.

La sécurité d'un réseau se situe à la fois au niveau physique et au niveau logiciel.

## **PRINCIPAUX OBJECTIFS DE LA SECURITE INFORMATIQUE**

Les objectifs principaux de la sécurité des systèmes d'information sont entre autres :

- la confidentialité : seules les personnes autorisées ont accès aux informations qui leur sont destinées ;
- l'intégrité : les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante.
- la disponibilité : Le système doit fonctionner sans faille durant les plages d'utilisation prévues et garantir l'accès aux services et ressources installées avec le temps de réponse attendu.
- la traçabilité (Preuve) : garantie que les accès et tentatives d'accès aux éléments considérés sont tracés et que ces traces sont conservées et exploitables.
- l'authentification : L'identification des utilisateurs est fondamentale pour gérer les accès aux espaces de travail pertinents et maintenir la confiance dans les relations d'échange.
- la non-répudiation : Aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées, et aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.

## **QUELQUES ATTAQUES ET TECHNIQUES DE PROTECTION**

S'il existe de multiples menaces qui pèsent sur un réseau d'entreprise, l'attaque d'un réseau commence traditionnellement par un balayage de ses ports, afin de déterminer lesquels sont ouverts. Cette opération est généralement faite par celui qu'on désigne par un hacker : c'est-à-dire quelqu'un qui tente d'accéder à votre réseau sans y être autorisé.

### **Quelques attaques connues sont :**

- le sniffing qui consiste à écouter le réseau ;
- l'usurpation d'adresse IP qui consiste à générer des paquets qui semblent provenir d'un poste accrédité pour la cible de manière frauduleuse ;
- The man-in-the-middle où le pirate joue l'intermédiaire entre le client et le serveur et peut récolter des informations confidentielles ou altérer une information avant de la transmettre.

### **Quelques techniques de protection**

Il existe de nombreuses techniques permettant de réduire l'impact des menaces qui pèsent sur votre système d'information. L'une des techniques les plus utilisées est le chiffrement de données que nous ne détaillerons pas dans cette activité qui a pour objectif de vous initier aux concepts de la sécurité réseau sous Linux.

Le chiffrement consiste de façon générale à rendre illisible les données par toute autre personne que celle à qui ces informations sont destinées grâce à un algorithme de chiffrement. Ce n'est qu'à l'aide d'une clé que le destinataire peut déchiffrer le message.

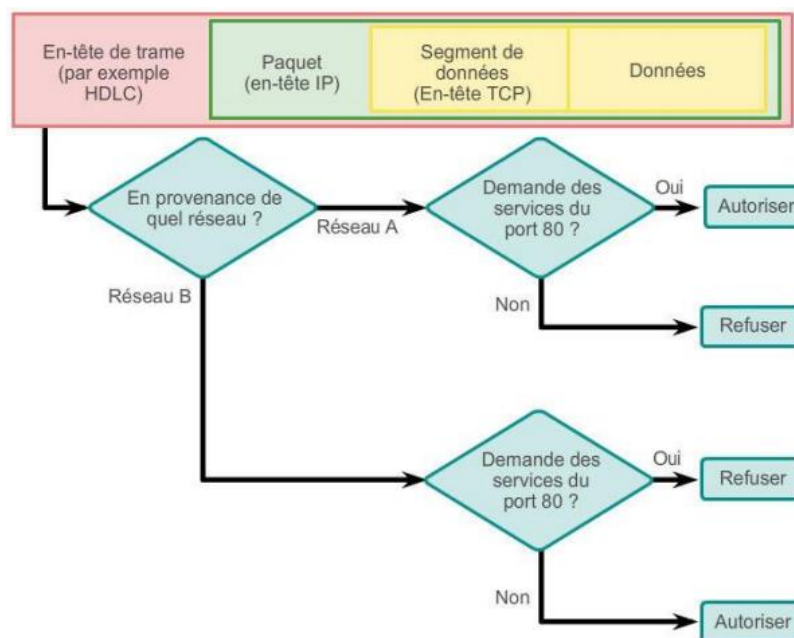
Si les techniques de chiffrement permettent de sécuriser la transmission des données privées sur un réseau public, elles ne peuvent pas vous protéger contre le trafic indésirable sur votre réseau.

Imaginez qu'un hacker arrive à connaître l'adresse IP du serveur de base de données de l'entreprise qui contient toutes les données sensibles de votre entreprise. Il envoie ensuite un flux continu de paquets ou exécute du code sur le serveur au point de mettre le serveur hors-service.

Pour anticiper ces problèmes et protéger vos systèmes contre le trafic inutile ou hostile, il existe ce qu'on appelle les pare-feu ou firewall en anglais.

Le **principe d'un pare-feu** est de ne laisser entrer que les types de trafic autorisés. Le but principal du pare-feu est de constituer un périmètre de sécurité autour du réseau privé. Un pare-feu doit être configuré de manière à ne laisser ouvert que les ports qui correspondent à un service utilisé sur le réseau privé et devant être accessible depuis l'extérieur.

Le filtrage de paquets est le mécanisme de base dont se sert un pare-feu. Encore appelé filtrage statique des paquets, ce mécanisme contrôle l'accès à un réseau en analysant les paquets qui entrent et sortent, puis les acceptent ou les rejettent selon des critères spécifiques, tels que l'adresse IP source, les adresses IP de destination et le protocole transporté dans le paquet.



Les ports laissés ouverts inutilement sont l'un des moyens les plus courants pour accéder frauduleusement à un réseau. C'est la raison pour laquelle il est indispensable que les administrateurs du réseau et du système surveillent avec soin quels sont les ports ouverts sur le pare-feu.

## **NOTION DE SERVICES ET DE PORTS**

Chaque service dans un environnement IP dispose d'un ou plusieurs numéros de ports qui lui sont réservés. Par exemple, un serveur Web utilise le port 80.

Q1. En vous servant du fichier `/etc/services` et la page web <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml> remplissez le tableau suivant :

Service	Numéro de port
FTP	
Telnet	
SMTP	
TFTP	
HTTP	
POP3	
HTTPS	
MYSQL	
SSH	
SIP	

Il existe de nombreux outils de sécurité, de vérification, de tests. L'outil nmap en fait partie. Il se définit comme un outil d'exploration réseau et d'audit de sécurité. Il permet de tester les connexions réseaux d'une machine donnée et de retourner un grand nombre d'informations.

Q2. Comment installer l'outil nmap ?

- Vérifiez la présence du logiciel nmap sur votre machine à l'aide du gestionnaire de paquets apt
- Si le logiciel n'est pas installé, procédez à son installation
- Vérifiez à nouveau que le logiciel nmap est installé avec `dpkg` cette fois-ci.

Q3. Comment effectuer un scan de port sur sa propre machine ?

- Services utilisant TCP  
`nmap -sS monip`
- Services utilisant UDP  
`nmap -sU monip`

Q4. Comment scanner un hôte précis dans le réseau ?

- Dans un premier temps, nous allons découvrir tous les hôtes disponibles sur notre réseau.

`nmap -sP adresse_reseau/masque_cidr`

- b. Nous allons vérifier si le port 22 est ouvert sur une machine du réseau

```
nmap -sUS -p 22 ip
```

### **INTERDIRE OU AUTORISER L'ACCES A UN SERVICE AVEC HOSTS.DENY ET HOSTS.ALLOW**

Q4. Comment interdire l'accès au service ssh à la machine ayant pour adresse IP x.x.x.x en utilisant les fichiers /etc/hosts.allow et /etc/hosts.deny ?

Afin de déterminer si un ordinateur client est autorisé à se connecter à un service, les enveloppeurs TCP référencent les deux fichiers suivants, couramment appelés fichiers d'accès des hôtes : /etc/hosts.allow et /etc/hosts.deny

La syntaxe est commune : daemon\_list : client\_list [:options]

daemon\_list : liste des exécutables (PAS DES SERVICES) séparés par des virgules. Vous pouvez mettre ALL pour spécifier tous les services. Si vous disposez de plusieurs interfaces réseau on peut utiliser la syntaxe avec @ : service@ip.

Pour avoir la liste des services utilisant les enveloppeurs TCP ou tcp\_wrappers, vous pouvez utiliser la commande suivante : strings -f <binaire> | grep hosts\_access (remplacer binaire par /usr/sbin/\*

client\_list : clients autorisés ou interdits pour ce service. On peut spécifier l'adresse IP, le nom, le masque de réseau, le nom du réseau, etc.

La liste des clients admet une syntaxe avancée :

ALL : correspondance systématique.

EXCEPT : permet d'exclure certains hôtes.

- a. A partir de ces informations, on vous demande d'écrire la règle dans le fichier /etc/hosts.deny permettant d'interdire l'accès au serveur ssh au client ayant pour adresse IP x.x.x.x
- b. Utilisez cette fois-ci le fichier /etc/hosts.allow pour autoriser l'accès à la même machine bloquée dans le fichier /etc/hosts.deny.  
Tirez une conclusion sur l'ordre de lecture des fichiers /etc/hosts.deny et /etc/hosts.allow

## **Netfilter et IPTABLES ([source](#))**

Le noyau Linux contient le logiciel NetFilter qui peut faire office de pare-feu. Il est paramétré et contrôlé par la commande iptables que nous allons étudier dans cette partie.

Le fonctionnement de IPTABLES est très simple et peut se résumer à : « des tables contenant des chaînes (chain) ; sur ces chaînes on peut appliquer des actions (target). »

La syntaxe de base de iptables est :

Iptables -A chaine -j cible

### **Découverte de deux (2) tables très importantes de iptables et des chaînes**

- Table 1 : filter

C'est la table par défaut, classée en 3 chaînes : INPUT (entrant), OUTPUT (sortant), et FORWARD (redirection)

Q1. Comment afficher la liste des règles de la table filter avec iptables ? (options -t et -L)

iptables -t table --line-numbers -L

- Table 2 : nat

C'est la table utilisée pour rediriger le trafic vers une autre machine. Elle possède les chaînes : PREROUTING (routage entrant), POSTROUTING (routage sortant), INPUT (entrant) et OUTPUT (sortant)

Q2. Comment afficher la liste des règles de la table nat ?

### **Quelques arguments très utilisés à connaître :**

-t : préciser la table

-A : indiquer la chaîne

-j : demander une action (accepter : ACCEPT, bloquer sans envoyer de message : DROP, Bloquer avec un message : REJECT)

-i : interface réseau de la connexion entrante

-o : interface réseau de la connexion sortante

-p : protocole

-s : ip source

-d : ip de destination

--sport : ports source

--dport : ports de destination

-F : vider une chaîne de ses règles

-D : supprimer une règle

Q3. Comment utiliser iptables pour interdire toute communication entrante avec une machine dont on connaît l'adresse IP ?

iptables -t filter -A INPUT -j DROP ou

iptables -t filter -A INPUT -j REJECT

Q4. Comment supprimer cette règle qu'on vient d'ajouter à la table FILTER ?

- a. Identifier le numéro de la règle  
iptables -t filter --line-numbers -L
- b. Procéder à la suppression  
Iptables -t filter -D INPUT 1

Q5. Sachant que le serveur SSH écoute sur le port 22 et utilise le protocole tcp, comment interdire l'accès à notre serveur ssh depuis la machine ayant l'adresse ip x.x.x.x

iptables -t filter -A INPUT -s x.x.x.x -p tcp --dport 22 -j REJECT

Q6. Comment supprimer toutes les règles ajoutées dans les tables filter et nat ?

iptables -t filter -F

iptables -t nat -F

## **Transformer sa machine Linux en routeur**

### **Partie 1** : Configuration d'un point d'accès

Vous avez à votre disposition les équipements suivants :

- Une machine Linux ayant une carte réseau ethernet

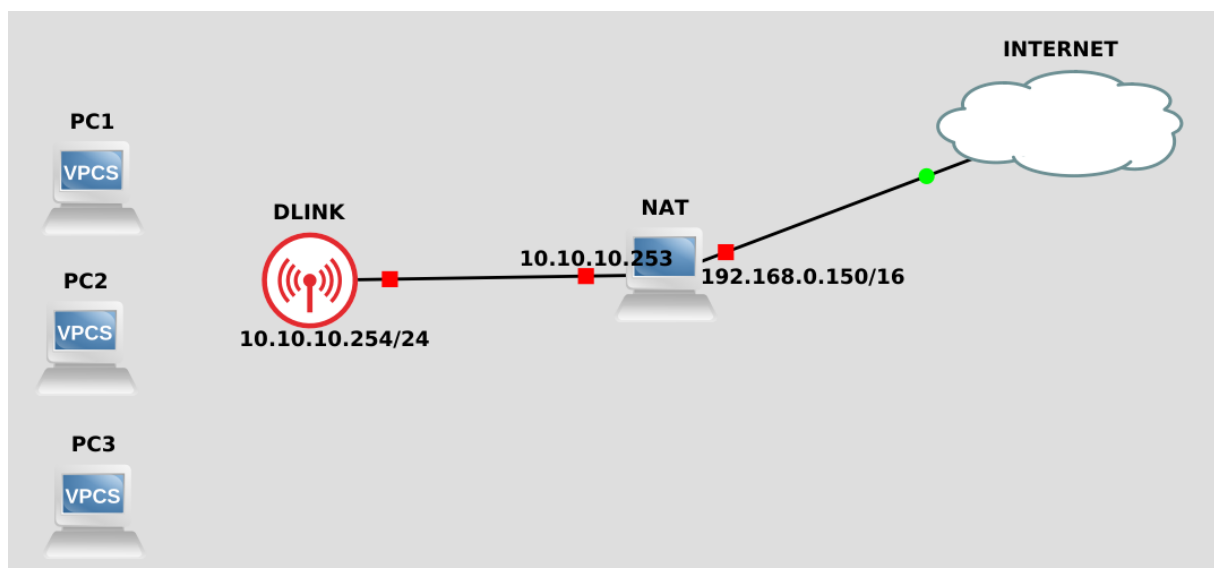
- Une carte ethernet USB afin que votre machine ait 2 cartes ethernet
- Un câble UTP
- Un point d'accès DLINK qu'on vient de réinitialiser. On vous demande de le configurer en suivant les étapes suivantes :
  - o A l'aide d'un câble UTP, reliez votre ordinateur au point d'accès
  - o Identifiez le réseau par défaut du point d'accès et donner à votre carte ethernet une adresse IP dans le même réseau (en utilisant Ubuntu et Windows)
  - o Connectez-vous à l'interface d'administration graphique du point d'accès en saisissant son adresse IP dans un navigateur
  - o Définir un SSID de votre choix (nom du réseau Wifi)
  - o L'adresse du point d'accès doit être 10.10.10.254
  - o Activez le service DHCP sur le point d'accès avec des adresses dans la plage 10.10.10.1 – 10.10.10.250

## Partie 2 : Transformer la machine Linux en routeur

### **2.1 AP (Point d'Accès) configuré en tant que serveur DHCP**

Sur la machine Ubuntu, on vous demande de :

- Brancher la carte réseau ethernet USB
- Faire les branchements nécessaires pour obtenir l'architecture suivante :



- Modifier le fichier `/etc/sysctl.conf`
- Chercher la ligne `"#net.ipv4.ip_forward = 1"` et la décommenter (supprimer le `#` devant celle-ci).
- Exécuter la commande suivante pour prendre en compte les modifications `sysctl -p /etc/sysctl.conf`
- Mettre en place le mécanisme NAT afin que les machines se connectant au réseau 10.10.10.0/24 puissent aller sur internet en passant par la passerelle 192.168.0.150 qui a accès à internet et fait office de Routeur NAT.



```
iptables -t nat -A POSTROUTING -o interface_connectee_a_internet -j MASQUERADE
```

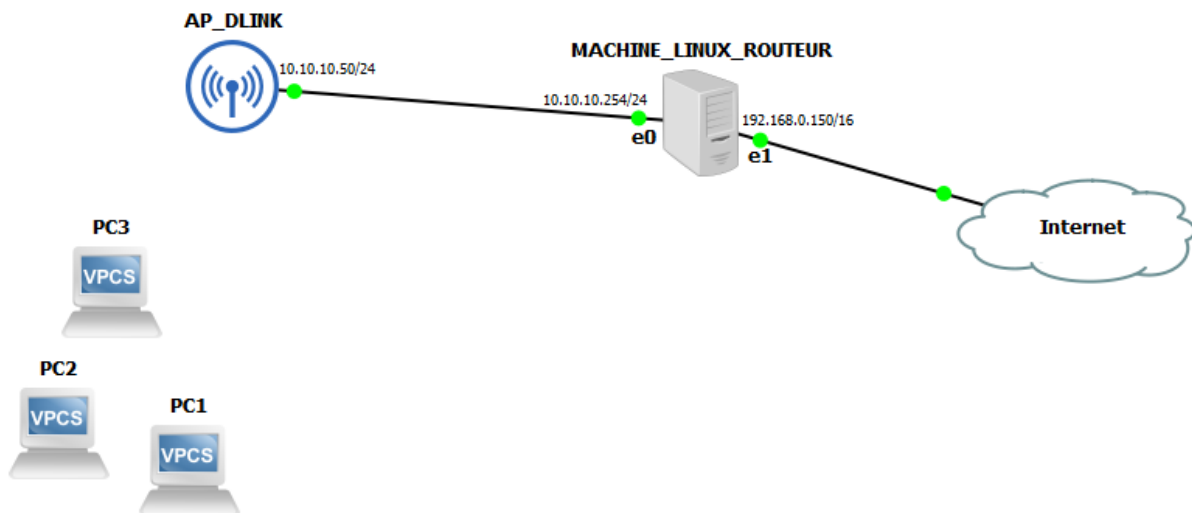
- Testez la communication vers une machine du réseau 192.168.0.0/16 à l'aide de l'utilitaire ping
- Tester la communication vers internet avec l'utilitaire ping vers un site web (exemple : ping google.sn)
- Fixer les éléments TCP/IP à une machine connectée au réseau WIFI du point d'accès en lui donnant comme passerelle l'adresse du routeur NAT et comme serveur DNS 8.8.8.8)
- Utilisez Wireshark sur la machine cliente et la machine serveur pour démontrer qu'il y a effectivement eu une translation d'adresse (NAT)

Q7. Comment utiliser iptables pour interdire aux clients connectés au réseau privé d'accéder à un site spécifique ?

- Faire un ping vers le site rtn.sn
  - Récupérer l'adresse IP publique du site
  - Mettre en place les deux règles suivantes qui empêchent toute communication entrante et sortante depuis et vers l'adresse IP du site identifié
- ```
iptables -A FORWARD -s 147.135.254.214 -j DROP  
iptables -A FORWARD -d 147.135.254.214 -j DROP
```

## 2.2 Machine Linux configurée comme serveur DHCP et DHCP désactivé sur le point d'accès

Soit l'architecture suivante à réaliser :



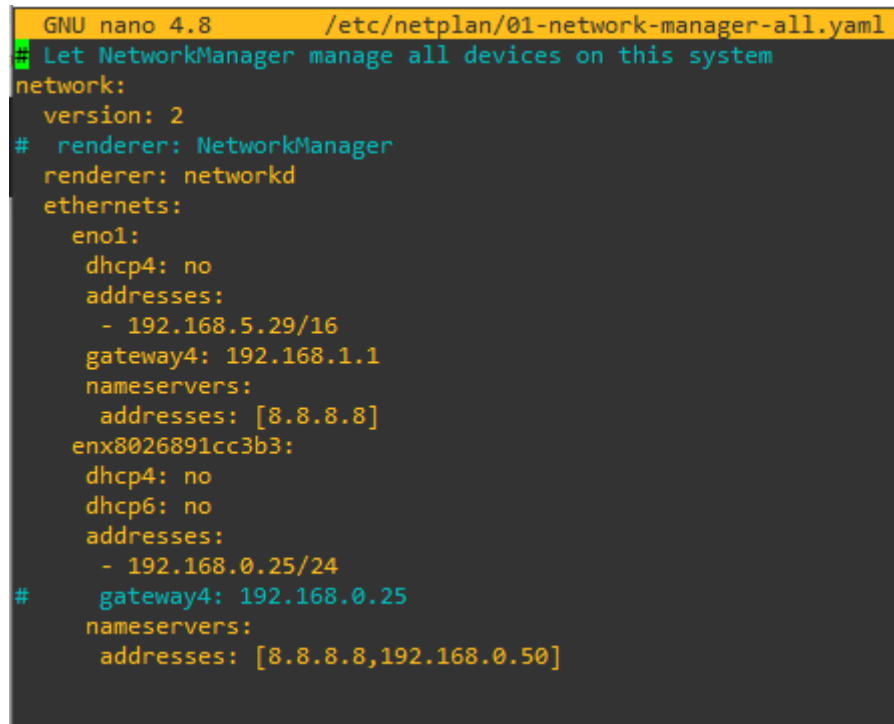
### Matériel :

- Une (1) machine Ubuntu 18
- Une (1) carte ethernet USB
- Un (1) point d'accès WIFI

- Deux (2) câbles UTP

Etape 1 : Fixer les éléments TCP/IP du point d'accès et des deux interfaces réseau en utilisant netplan et le gestionnaire networkd

*Inspirez-vous des exemples ci-dessous pour fixer les adresses conformément à l'architecture prévue.*



```
GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
#  renderer: NetworkManager
  renderer: networkd
  ethernets:
    eno1:
      dhcp4: no
      addresses:
        - 192.168.5.29/16
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8]
    enx8026891cc3b3:
      dhcp4: no
      dhcp6: no
      addresses:
        - 192.168.0.25/24
#      gateway4: 192.168.0.25
      nameservers:
        addresses: [8.8.8.8, 192.168.0.50]
```

*Exécuter la commande « netplan apply » pour prendre en compte les modifications*

*L'image ci-dessous représente la configuration statique des éléments TCP du point d'accès.*

LAN SETUP

configure this device more easily when your network using TCP/IP protocol. You can enter the device name of the AP into your web browser to access the instead of IP address for configuration. Recommend to change the device name if there're more than one D-Link devices within the subnet.

Save Settings Don't Save Settings

DEVICE NAME

Device Name allows you to configure this device more easily. You can enter "http://"device name into your web browser instead of IP address for configuration. (Default: http://dlinkap)

Device Name :

LAN IPV4 CONNECTION TYPE

Choose the IPv4 mode to be used by the Access Point

MY LAN Connection is :

STATIC IP ADDRESS LAN CONNECTION TYPE:

Enter the IPv4 Address information.

IP Address :

Subnet Mask :

Gateway Address :

Primary DNS Server :

Secondary DNS Server :

## Etape 2 : Installer le paquet isc-dhcp-server et faire les configurations nécessaires

- Définir l'interface qui va recevoir les requêtes DHCP

```

GNU nano 4.8 /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enx8026891cc3b3"
INTERFACESv6=""

```

- Mettre en place le DHCP proprement dit (les adresses IP sont à adapter à votre TP)

```

GNU nano 4.8 /etc/dhcp/dhcpd.conf
#subnet 10.254.239.32 netmask 255.255.255.224 {
#   range dynamic-bootp 10.254.239.40 10.254.239.60;
#   option broadcast-address 10.254.239.31;
#   option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 192.168.0.0 netmask 255.255.255.0{
    range 192.168.0.51 192.168.0.254;
    option domain-name-servers 8.8.8.8;
#   option domain-name "internal.example.org";
#   option subnet-mask 255.255.255.224;
    option routers 192.168.5.29;
#   option broadcast-address 10.5.5.31;
    default-lease-time 8600;
    max-lease-time 7200;
}

```

- Démarrer le serveur DHCP et s'assurer que tout fonctionne (service isc-dhcp-server start)

```

root@bessan-ubuntu:/home/bessan# service isc-dhcp-server status
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vend
   Active: active (running) since Thu 2021-06-24 19:22:55 GMT; 13min ago

```

**Etape 3 : Configuration du mécanisme NAT Dynamique de type MASQUERADE (une seule adresse IP publique)**

**iptables -t nat -A POSTROUTING -o eno1 -j MASQUERADE**

**Etape : Configuration du mécanisme Destination NAT avec la redirection de port et mise en évidence du PAT**

**iptables -t nat -A PREROUTING -d 192.168.168.5.29 -i eno1 -p tcp --dport 22 -j DNAT -to-destination 192.168.0.57:22**

**Travail à faire à la maison :**

Toutes les règles définies ne sont pas permanentes et vont s'effacer dès que vous allez redémarrer la machine. On vous demande d'effectuer les recherches suivantes :

- Comment rendre les règles iptables persistentes ?
- Existe-t-il d'autres pare-feu sous Ubuntu ? Si oui, lesquels et comment fonctionnent-ils ? (ufw, apparmor)
- Comment gérer la définition d'éléments TCP/IP manuellement pour se connecter à un réseau sans fil ?

