GALLAND Cyprien VATIN Clément

TP 5 : Services réseau

Exercice 1 : Adressage IP (Rappels)

On administre le réseau interne 172.16.0.0/23 d'une entreprise, et devons gérer un parc de 254 machines réparties en 7 sous-réseaux.

Voici le tableau de répartition des IP :

Sous-réseau	Nb de machines	Adresse du sous-réseau	Adresse de broadcast	Adresse de la 1ere machine	Adresse de la dernière machine
1	38	172.16.0.0/26	172.16.0.63/26	172.16.0.1/26	172.16.0.38/26
2	33	172.16.0.64/26	172.16.0.127/26	172.16.0.65/26	172.16.0.97/26
3	52	172.16.0.128/26	172.16.0.191/26	172.16.0.129/26	172.16.0.180/26
4	35	172.16.0.192/26	172.16.0.255/26	172.16.0.193/26	172.16.0.227/26
5	34	172.16.1.0/26	172.16.1.63/26	172.16.1.1/26	172.16.1.34/26
6	37	172.16.1.64/26	172.16.1.127/26	172.16.1.65/26	172.16.1.101/26
7	25	172.16.1.128/26	172.16.1.191/26	172.16.1.129/26	172.16.1.153/26

Exercice 2 : Préparation de l'environnement

Dans ce TP, Il s'agit de mettre en place un réseau rudimentaire constitué de seulement deux machines : un serveur et un client :

- le serveur a une connexion Internet, notamment pour télécharger les paquets nécessaires à l'installation des serveurs, et sert de passerelle au client ;
    - les deux machines appartiennent à un réseau local, tpadmin.local, ayant pour adresse 192.168.100.0/24
    - le client a accès à Internet uniquement via le serveur; il dispose d'une interface réseau qui recevra son adresse IP du serveur DHCP.

1) VM éteintes, on configure les 2 VMs comme il faut, c'est à dire en donnant à la VM serveur 2 cartes réseaux, l'une normale (NAT), et l'autre dédiée au réseau local (réseau interne), et à la VM client une seule carte réseau, donnant accès au réseau interne.

2) On démarre ensuite le serveur, puis on exécute les 2 commandes suivantes :

```
ip -4 a

ip a
```

La première de ces deux commandes renvoie les interfaces possédant une adresse IPv4, soit ici seulement deux interfaces réseau : L'adresse de loopback (c'est à dire celle de la machine elle même), et l'adresse du réseau externe. Pour afficher toutes les interfaces réseau, il faut la deuxième commande. En plus des deux interfaces déja citées, elle renvoie également une troisième interface, non configurée, et par défaut en IPv6 : Notre réseau local.

Exercice 3 : Installation du serveur DHCP

Un serveur DHCP permet aux ordinateurs clients d'obtenir automatiquement une configuration réseau (adresse IP, serveur DNS, passerelle par défaut…), pour une durée déterminée.

On va en mettre un en place pour configurer automatiquement l'interface réseau du client par le serveur.

1) On commence par installer le paquet isc-dhcp-server sur le serveur :

```
sudo apt install isc-dhcp-server
```

Ensuite, on exécute la commande :

```
systemctl status isc-dhcp-server
```

Or cette commande renvoie un message d'erreur. Pourquoi ? La raison est simple : le serveur n'étant pas encore configuré, il ne peut pas démarrer...

2) Pour commencer, le serveur à besoin d'une IP statique. Pour attribuer celle ci de manière permanente, on passe par netplan :

```
sudo nano /etc/netplan/51-cloud-init.yaml #On crée un nouveau fichier netplan, de numér
```

Dans ce fichier, on recopie les lignes suivantes :

```
network :

    version : 2

    ethernets :

        enp0s8 :

            addresses :

                - 192.168.100.1/24
```

ATTENTION : Netplan n'accepte pas les tabulation, il faut taper 4 espaces à chaque fois.

On peut ensuite retourner en ligne de commande :

```
sudo netplan try

sudo netplan apply
```

On a configuré le serveur DHCP en lui affectant l'IP 192.168.100.1 !

3) On va maintenant achever de configurer le serveur, via le fichier /etc/dhcp/dhcpd.

Après en avoir au préalable fait une copie, on peut remplacer son contenu par :

```
default-lease-time 120;

max-lease-time 600;

authoritative; # Indique qu'il s'agit du DHCP officiel du réseau

option broadcast-address 192.168.100.255

option domain-name "tpadmin.local"

subnet 192.168.100.0 { # Configuration du sous réseau 192.168.100.0

range 192.168.100.100 192.168.100.240 # Plage d'adresses distribuables

option routers 192.168.100.1 #Le serveur est passerelle par défaut

option domain-name-servers 192.168.100.1 # Le serveur fait aussi office de DNS

}
```

4) On édite maintenant le fichier /ect/default/isc-dhcp-server, en précisant que le serveur doit écouter sur

l'interface 192.168.100.1.

5) On valide maintenant le fichier de configuration :

```
dhcpd -t

sustemctl restart isc-dhcp-server #Redémarrage du serveur
```

Enfin un `ip -4 a` (qui affiche désormais les 3 interfaces) permet de vérifier que le serveur DHCP est bien actif.

6) On va pouvoir désormais s'occuper du client. La première chose à faire est de modifier le nom d'hôte (comme il s'agit d'un clone de serveur, son nom d'hôte est à ce jour... Serveur !). Après avoir désactivé l'interface réseau, on allume le client, puis on saisit les commandes suivantes :

```
hostnamectl set-hostname client

nano /etc/cloud/cloud.cfg.d/99_hostname.cfg
```

Dans ce dernier fichier, on rajoute simplement : `preserve_hostname: true`

Cela à pour effet de garder le nouveau nom d'hôte lors du prochain démarrage de la machine.

Cela étant fait, on éteint le client, et dans les configurations on lui active l'interface réseau (sans le lancer pour l'instant...)

7) Sur le serveur, on exécute la commande :

```
tail -f /var/log/syslog
```

Qui à pour effet d'afficher en continu les logs sur le serveur.

On allume ensuite le client, et on observe sur le serveur un certains nombres de message :

- DHCHPDISCOVER : indique que le serveur à découvert une nouvelle machine sur le réseau interne, et via quel port à eu lieu la découverte.
- DHCPOFFER : Le serveur propose une adresse à la machine découverte.
- DHCPREQUEST : le client demande à ce que cette adresse lui soit attribuée
- DHCPACK : le serveur accède à cette demande.

Les deux dernières lignes sont répétées régulièrement, selon un temps défini lors de la configuration du serveur DHCP : Les adresses sont en effet attribuées pour un temps défini : ce temps expiré, il faut les réattribuer.

On peut constater que le client reçoit l'adresse 192.168.100.100, qui est bien une adresse de la plage définie précédemment.

8) Le fichier /var/lib/dhcp/dhcpd.leases sur le serveur contient l'historique des diverses sessions de connexion du client sur le serveur (en fait, de toutes les machines ayant eu une session sur le serveur DHCP). La commande `dhcp-lease-list`, elle, affiche la liste des sessions de connexion en cours sur le serveur.

9) Un `ping 192.168.100.100` sur le serveur, et `ping 192.168.100.1` sur le client permettent de vérifier que les machines se voient bien.

10) Non traitée (il s'agit de modifier le serveur pour qu'il affecte l'adresse statiquement, cela se fait sans difficulté en bidouillant le netplan).

Exercice 4 : Donner un accès à internet au client

A ce stade, le client est juste une machine sur notre réseau local, et n'a aucun accès à Internet... Pour remédier à cette situation, on va se servir de la machine serveur (qui, elle, a un accès à Internet via son autre carte réseau) comme d'une passerelle.

1) On autorise sur le serveur l'IP forwarding, désactivée par défaut : Il faut aller dans le fichier /etc/susctl.conf, et dé-commenter la ligne *net.ipv4.ip_forward=1*.

On exécute ensuite les commandes :

```
sudo sysctl -p /etc/sysctl.conf #permet la prise en compte immédiate des changements

sysctl net.ipv4.ip_forward # Renvoie 1, preuve que le changement à été pris en compte
```

2) On autorise ensuite le masquerading (traduction d'adresse source) :

```
sudo iptables --table nat --append POSTROUTING --out-interface enp0s3 -j MASQUERADE
```

On vérifie ensuite qu'il est possible de pinger n'importe quelle IP depuis le client :

`ping 1.1.1.1`

Cependant, il demeure impossible de naviguer convenablement sur le net : il faut en effet configurer un serveur DNS sur le client.

Exercice 5: Configuration d'un serveur DNS sur le client

DNS (Domain Name Server) : solution qui permet de faire le lien entre adresse ip et nom de domaine.

Il est commun d'utiliser un serveur DNS privé, interne à l'organisation, afin de pouvoir résoudre les noms des machines locales. Pour les requêtes extérieures, le serveur DNS privé passe la main à un DNS externe.

Il existe de nombreux serveurs DNS. Celui que nous allons utiliser est Bind9.

1) On commence par installer bind9 sur le serveur :

```
sudo apt install bind9 dnsutils
```

2) On va configurer bind comme un serveur cache, c'est à dire un serveur qui met en cache les réponses de serveurs externes, à qui il transmet les requêtes de résolution de nom.

Attention Le programme installé ne s'appelle pas bind9, mais named... On modifie son fichier de configuration :

```
sudo nano /etc/bind/named.conf.options
```

Dans ce fichier, on dé-commente la partie *forwarders*, et on remplace 0.0.0.0 par 1.1.1.1 (cloudflare) et 8.8.8.8 (google). Puis on redémarre bind9 :

```
sudo service bind9 restart
```

3) On va ensuite sur le client :

```
ping www.google.fr
```

Et cette fois, le ping fonctionne !

4) Toujours sur le client, on installe un navigateur en mode texte, nommé lynx :

```
sudo apt install lynx
```

On tape ensuite en ligne de commande :

```
lynx fr.wikipedia.org
```

Et on se retrouve à naviguer sur wikipédia. Et franchement, l'interface graphique nous manque beaucoup !

Exercice 6 : Configuration du serveur DNS pour la zone tpadmin.local

L'intérêt d'un serveur DNS privé est principalement de pouvoir résoudre les noms des machines du réseau local. Pour l'instant, il est impossible de pinguer client depuis serveur et inversement.

1) Sur le serveur, on modifie le fichier /etc/bind/named.conf.local (`sudo nano /etc/bind/named.conf.local`, auquel on ajoute les lignes suivantes :

```
zone "tpadmin.local" {

    type master; // c'est un serveur maître file

    "/etc/bind/db.tpadmin.local"; // lien vers le fichier de définition de zone

};
```

2) On crée désormais une copie, nommée db.tpadmin.local, du fichier db.local. Il s'agit d'un fichier de configuration typique de DNS. Dans cette copie, on remplace localhost par tpadmin.local, et 127.0.0.1 par 192.168.100.1 (adresse IP du serveur).

La ligne root.tpadmin.local. indique en fait une adresse mail du responsable technique de cette zone, où le symbole @ est remplacé par un point. Attention également à ne pas oublier le point final, qui représente la racine DNS ;

Le champ serial doit être incrémenté à chaque modification du fichier. Généralement, on lui donne pour valeur la date suivie d'un numéro sur deux chiffres, ici 2019031901.

3) Maintenant que nous avons configuré notre fichier de zone, il reste à configurer le fichier de zone inverse, qui permet de convertir une adresse IP en nom.

```
sudo nano /etc/bind/named.conf.local
```

A la fin du fichier, on rajoute les lignes suivantes :

```
zone "100.168.192.in-addr.arpa" {

    type master;

    file "/etc/bind/db.192.168.100";

};
```

On crée ensuite, sur le modèle du fichier /etc/bind/db.127, le fichier /etc/bind/db.192.168.100 (pour ce faire, on effectue une copie de /etc/bind/db.127 dont on change le nom).

On modifie ce fichier comme le précédent (localhost devient tpadmin.local, et serial est incrémenté)

Sur la dernière ligne, on fait correspondre avec l'IP du serveur en ne marquant que la partie qui n'est pas dans le nom du fichier (ici, juste 1; Pour db.127, c'était 1.0.0)

4) On valide les fichiers de configuration :

```
 $ named-checkconf /etc/bind/named.conf.local

$ named-checkzone tpadmin.local /etc/bind/db.tpadmin.local

$ named-checkzone 100.168.192.in-addr.arpa /etc/bind/db.192.168.100
```

Si il n'y a pas d'erreurs, il renvoie un message OK aux deux dernières instructions.

5) On redémarre le serveur Bind9, et on ping les différentes machines du réseau : tout marche.