Administration système - TP 4

Auteurs :

BIARD Gauthier

Le 28/02/2020

Exercice 1. Adressage IP (rappels)

Vous administrez le réseau interne 172.16.0.0/23 d'une entreprise, et devez gérer un parc de 254 machines réparties en 7 sous-réseaux. La répartition des machines est la suivante :

- **Sous-réseau 1 : 38 machines**
- **Sous-réseau 2 : 33 machines**
- **Sous-réseau 3 : 52 machines**
- **Sous-réseau 4 : 35 machines**
- **Sous-réseau 5 : 34 machines**
- **Sous-réseau 6 : 37 machines**
- **Sous-réseau 7 : 25 machines Donnez, pour chaque sous-réseau, l'adresse de sous-réseau, l'adresse de broadcast (multidiffusion) ainsi que les adresses de la première et dernière machine configurées (précisez si vous utilisez du VLSM ou pas).**

Sous-réseau	Adresse de sous-réseau	Adresse de broacast	Première machine	Dernière machine
1 *(38 machines)*	172.16.0.0	172.16.0.63	172.16.0.1	172.16.0.62
2 *(33 machines)*	172.16.0.64	172.16.0.127	172.16.0.65	172.16.0.126
3 *(52 machines)*	172.16.0.128	172.16.0.191	172.16.0.129	172.16.0.190
4 *(35 machines)*	172.16.0.192	172.16.0.255	172.16.0.193	172.16.0.254
5 *(34 machines)*	172.16.1.0	172.16.1.63	172.16.1.1	172.16.1.62
6 *(37 machines)*	172.16.1.64	172.16.1.127	172.16.1.65	172.16.1.126
7 *(25 machines)*	172.16.1.128	172.16.1.159	172.16.1.129	172.16.1.158

> Pour les sous-réseaux 1 à 6, nous avons besoin de minimum 33 machines et maximum 52 machines. Or, pour l'adressage IP, les sous-réseau que nous pouvons créer doivent être des puissance de 2, c'est-

> à-dire, des sous-réseau soit de 32 machines soit de 64 machines. Donc pour les sous-réseaux 1 à 6, nous devons utiliser des sous-réseau de 64 machines (donc des adresses IP de masque /26). Cependant, pour le sous-réseau 7, nous n'avons que 25 machines, donc nous utilisons un sous-réseau de 32 machines, soit une adresse IP réseau de masque /27.

Exercice 2. Préparation de l'environnement

Dans ce TP nous allons 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 (on aurait pu choisir une autre adresse, sauf 192.168.1.0/24 qui est souvent réservé, par exemple par le FAI);**
- **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, utilisez les outils de configuration de VirtualBox pour mettre en place l'environnement décrit ci-dessus

2. Démarrez le serveur et vérifiez que les interfaces réseau sont bien présentes. A quoi correspond l'interface appelée lo ?

L'interface *lo* est l'interface de loopback. Il s'agit d'un réseau virtuel permettant de contacter la machine en local sans passer par une interface accessible de l'extérieur (de base, l'adresse associée est le *127.0.0.1*).

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. Ainsi, dans notre cas, l'interfaces réseau de client doit être configurée automatiquement par serveur.

1. Sur le serveur, installez le paquet isc-dhcp-server. La commande systemctl status isc-dhcp-server devrait vous indiquer que le serveur n'a pas réussi à démarrer, ce qui est normal puisqu'il n'est pas encore configuré (en particulier, il n'a pas encore d'adresses IP à distribuer).

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

2. Un serveur DHCP a besoin d'une IP statique. Attribuez de manière permanente l'adresse IP 192.168.100.1 à l'interface réseau du réseau interne. Vérifiez que la configuration est correcte.

On modifie le dossier de configuration */etc/netplan* :

```
sudo nano /etc/netplan/51-cloud-init.yaml #On crée un nouveau fichier netplan, de
numéro plus élevé que celui d'avant
network:
    version: 2
    renderer: NetworkManager (ou networkd)

    ethernets:
        enp0s8:
            addresses:
                - 192.168.100.1/24
```

> Nous pouvons ensuite vérifier à l'aide de la commande **ip a** si l'adressage est bon. **netplan** permet que l'adressage IP d'une machine soit permanant, ce qui n'est pas le cas avec la commande ifconfig.

> Nous ne pouvons pas utiliser de tabulations (il faut mettre 4espaces)

3. La configuration du serveur DHCP se fait via le fichier /etc/dhcp/dhcpd.conf. Renommez le fichier existant sous le nom dhcpd.conf.bak puis créez en un nouveau avec les informations suivantes :

```
default-lease-time 120;
max-lease-time 600;
authoritative; #DHCP officiel pour notre réseau
option broadcast-address 192.168.100.255; #informe les clients de l'adresse de
broadcast
option domain-name "tpadmin.local"; #tous les hôtes qui se connectent au #réseau
auront ce nom de domaine
subnet 192.168.100.0 netmask 255.255.255.0 { #configuration du sous-réseau
192.168.100.0
range 192.168.100.100 192.168.100.240; #pool d'adresses IP attribuables
option routers 192.168.100.1; #le serveur sert de passerelle par défaut
option domain-name-servers 192.168.100.1; #le serveur sert aussi de serveur DNS
}
```

A quoi correspondent les deux premières lignes ?

Les valeurs indiquées sur ces deux lignes sont faibles, afin que l'on puisse voir constituer quelques logs durant ce TP. Dans un environnement de production, elles sont beaucoup plus élevées!

Ces deux lignes définissent le temps entre chaque reload du serveur.

4. Editez le fichier /etc/default/isc-dhcp-server afin de spécifier l'interface sur laquelle le serveur doit écouter.

Le serveur doit écouter l'interface 192.168.100.1

5. Validez votre fichier de configuration avec la commande dhcpd -t puis redémarrez le serveur DHCP (avec la commande systemctl restart isc-dhcp-server) et vérifiez qu'il est actif.

```
dhcpd -t
sustemctl restart isc-dhcp-server #on redémarre le serveur dhcp
```

Avec la commande *ip a*, nous pouvons vérifier que le serveur dhcp est bien actif.

6. Passons au client. Si vous avez suivi le sujet du TP 1, le client a été créé en clonant la machine virtuelle du serveur. Par conséquent, son nom d'hôte est toujours serveur. Nous allons remédier à cela. Pour l'instant, vérifiez que la carte réseau du client est désactivée, puis démarrez le client. Pour modifier le nom de la machine, saisissez la commande hostnamectl set-hostname client.

Dans les versions récentes, Ubuntu installe d'office le paquet cloud-init lors de la configuration du système. Ce paquet permet la configuration de machines via un script dans le cloud, et a parfois des effets de bord fâcheux; en particulier, il supprimera le nom qu'on vient de donner à notre VM au prochain redémarrage pour lui redonner son ancien nom. Pour éviter cela, créez le fichier /etc/cloud/cloud.cfg.d/99_hostname.cfg dans lequel vous ajouterez simplement preserve_hostname: true

Tout d'abord, nous désactivons l'iterface réseau puis, après avoir allumé le client, nous modifions le nom du client et les paramètre avec les commandes suivantes :

```
hostnamectl set-hostname client
nano /etc/cloud/cloud.cfg.d/99_hostname.cfg
```

> Dans le fichier, nous rajoutons : *preserve_hostname: true*. Avec cela, le nom est conserver au démarrage suivant. Nous éteignons le client, puis nous actions l'interface réseau.

7. La commande tail -f /var/log/syslog affiche de manière continue les dernières lignes du fichier de log du système (dès qu'une nouvelle ligne est écrite à la fin du fichier, elle est affichée à l'écran). Lancez cette commande sur le serveur, puis activez la carte réseau du client et observez les logs sur le serveur. Expliquez à quoi correspondent les messages DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK. Vérifiez que le client reçoit bien une adresse IP de la plage spécifiée précédemment.

Sur le serveur, nous affichons les logs en continue sur le serveur avec la commande suivante :

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

Nous allumons maintenant le client et observons différents messages dur le serveur :

- 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 lignes *DHCPREQUEST* et *DHCPACK* sont répétées régulièrement, selon un temps défini lors de la configuration du serveur DHCP : les adresses sont attribuées pour un temps défini et lorsque 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. Que contient le fichier /var/lib/dhcp/dhcpd.leases sur le serveur, et qu'afficle la commande dhcp-lease-list ?

Ce fichier contient l'historique des diverses sessions de connexion du client sur le serveur (de toutes les machines). La commande dhcp-lease-list affiche la liste des sessions de connexion en cours sur le serveur.

9. Vérifiez que les deux machines se « voient » via leur adresse IP, à l'aide de la commande ping.

En faisant un ping, les deux machines se voient bien.

10. Modifiez la configuration du serveur pour que l'interface réseau du client reçoive l'IP statique 192.168.100.20 :

```
deny unknown-clients; #empêche l'attribution d'une adresse IP à une
                      #station dont l'adresse MAC est inconnue du serveur
host client1 {
hardware ethernet XX:XX:XX:XX:XX:XX; #remplacer par l'adresse MAC
fixed-address 192.168.100.20;
}
```

Vérifiez que la nouvelle configuration a bien été appliquée sur le client (éventuellement, désactivez puis réactivez l'interface réseau pour forcer le renouvellement du bail DHCP, ou utilisez la commande dhclient -v).

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. La première chose à faire est d'autoriser l'IP forwarding sur le serveur (désactivé par défaut, étant donné que la plupart des utilisateurs n'en ont pas besoin). Pour cela, il suffit de décommenter la ligne net.ipv4.ip_forward=1 dans le fichier /etc/sysctl.conf. Pour que les changements soient pris en compte immédiatement, il faut saisir la commande sudo sysctl -p /etc/sysctl.conf.

Vérifiez avec la commande sysctl net.ipv4.ip_forward que la nouvelle valeur a bien été prise en compte.

```
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. Ensuite, il faut autoriser la traduction d'adresse source (masquerading) en ajoutant la règle iptables suivante :

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

Vérifiez à présent que vous arrivez à « pinguer » une adresse IP (par exemple 1.1.1.1 depuis le client.

A ce stade, le client a désormais accès à Internet, mais il sera difficile de surfer : par exemple, il est même impossible de pinguer www.google.com. C'est parce que nous n'avons pas encore configuré de serveur DNS pour le client.

Il faut maintenant configurer un serveur DNS.

Exercice 5. Installation du serveur DNS

1. Sur le serveur, commencez par installer Bind9, puis assurez-vous que le service est bien actif

```
sudo apt install bind9 dnsutils
```

2. A ce stade, Bind n'est pas configuré et ne fait donc pas grand chose. L'une des manières les simples de le configurer est d'en faire un serveur cache : il ne fait rien à part mettre en cache les réponses de

serveurs externes à qui il transmet la requête de résolution de nom.

Le binaire (= programme) installé avec le paquet bind9 ne s'appelle ni bind ni bind9 mais named...

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

On décommente la partie *forwarders* et on remplace donc l'adresse 0.0.0.0 par 1.1.1.1 et 8.8.8.8 qui sont les DNS de cloudflare et google. Puis nous redémarrons bind9 :

3. Sur le client, retentez un ping sur www.google.fr. Cette fois ça devrait marcher! On valide ainsi la configuration du DHCP effectuée précédemment, puisque c'est grâce à elle que le client a trouvé son serveur DNS.

Cette fois, le ping fonctionne.

4. Sur le client, installez le navigateur en mode texte lynx et essayez de surfer sur fr.wikipedia.org (bienvenue dans le passé...)

```
sudo apt install lynx
```

Afin de naviguer, nous exécutons :

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

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. Modifiez le fichier /etc/bind/named.conf.local et ajoutez 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. Créez une copie appelée db.tpadmin.local du fichier db.local. Ce fichier est un fichier configuration typique de DNS, constitué d'enregistrements DNS (cf. cours). Commencez par remplacer localhost par tpadmin.local, et l'adresse 127.0.0.1 par l'adresse IP du serveur.

L'adresse IP du serveur est 192.168.100.1.

> 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.

Commencez par rajouter les lignes suivantes à la fin du fichier named.conf.local :

```
zone "100.168.192.in-addr.arpa" {
type master;
file "/etc/bind/db.192.168.100";
};
```

Créez ensuite le fichier db.192.168.100 à partir du fichier db.127, et modifiez le de la même manière que le fichier de zone. Sur la dernière ligne, faites correspondre l'adresse IP avec celle du serveur (Attention, il y a un petit piège !).

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. Utilisez les utilitaires named-checkconf et named-checkzone pour valider vos fichiers de configuration :

```
$ named-checkconf 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
```

5. Redémarrer le serveur Bind9. Vous devriez maintenant être en mesure de "pinguer" les différentes machines du réseau.

Tout fonctionne.