Rapport SAE 2.03

Installation de services réseau

BERLUREAU Nathan
IUT DU LIMOUSIN | Département Informatique

Année 1 Groupe 2B

Table des matières

Glossaire :

Pour ce qui est du glossaire, nous avons mis ces éléments qui nous semblaient importants : CIDR définit la partie réseau et le nombre de machines pouvant être présentes sur ce réseau

DHCP (Dynamic Host Configuration Protocol) est un un protocole de la couche application (7), utilisé pour attribuer automatiquement les adresses IP et les autres paramètres de configuration réseau aux dispositifs connectés à un réseau informatique.

DNS (Domain Name System) utilisé à la couche 7 du modèle OSI, est un système essentiel sur Internet qui permet de traduire les noms de domaine compréhensibles par les humains en adresses IP, qui sont les identifiants numériques des dispositifs connectés au réseau. Il sert principalement à faciliter la communication entre les utilisateurs et les serveurs sur Internet.

FTP (File Transfer Protocol) est un protocole réseau standard utilisé pour transférer des fichiers entre un client et un serveur sur un réseau informatique. Il permet le téléchargement et le téléversement de fichiers, ainsi que d'autres opérations de gestion de fichiers.

Kathara est une plateforme de virtualisation réseau utilisée dans le domaine des réseaux informatiques. Elle permet de créer des topologies de réseau complexes et de simuler des environnements réseau pour des fins d'apprentissage, de développement et de test.

PING permet de vérifier la connectivité entre deux machines.

SFTP (SSH File Transfer Protocol) est une version sécurisée du protocole FTP qui utilise le protocole SSH pour fournir un canal sécurisé. Il assure une protection accrue lors du transfert de fichiers en cryptant les données échangées entre le client et le serveur.

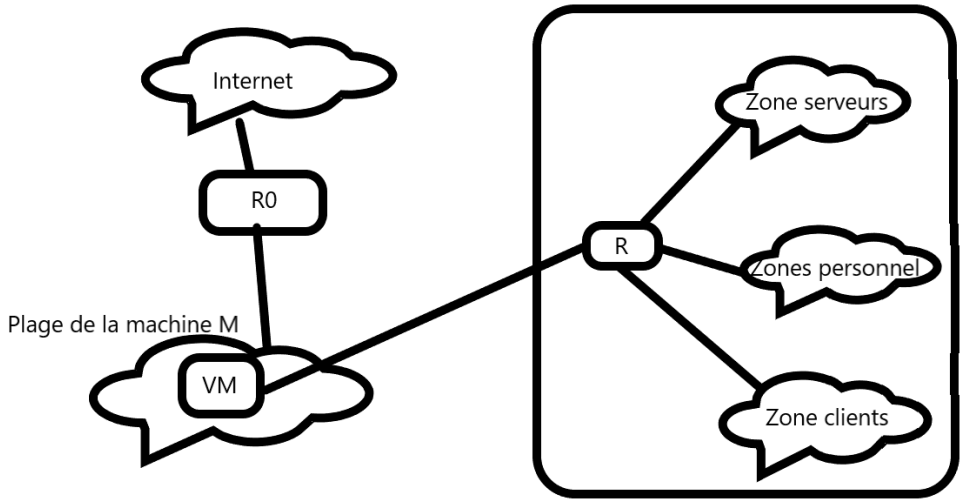
SSH (Secure Shell), est un protocole de communication (utilisé à la couche 7 du modèle OSI), sécurisé largement utilisé pour se connecter à des systèmes distants et exécuter des commandes à distance.

Wireshark est un logiciel d'analyse de protocole réseau il permet de capturer, d'analyser et de visualiser le trafic réseau en temps réel.

Rappel du sujet :

Dans cette SAE, il nous était demandé de configurer une infrastructure interne privée.
Il y a avait 3 zones à mettre en place, une « zone personnel » pour les développeurs et une « zone clients » qui permettra au futurs clients l’accès au réseau lorsqu’ils visiteront l’entreprise. On devait donc diviser la plage d’adresse qui nous était donné pour que au moins 289 adresses machines soient utilisé dans la zone personnel et au moins 700 adresses machines pour la zone client. Bien sur l’entreprise demande en plus 2 plages d’adresses supplémentaires pour s’occuper de connecter les routeurs entre eux et de configurer les machines de la zones serveurs.

On nous donnes ensuite un schéma qui représente l’architecture souhaitant être installé :



Ensuite il nous était demandé plusieurs choses en fonctions des différentes zones de travail,
- Dans la zone serveurs on devait configurer une machine SF qui possède la première adresse de la zone et son routeur R_S qui permet de relier cette zone à R (et donc au autre zone et à internet).

- Dans la zone personnel, 3 machines devront être configuré, Pca Pcb et R_P, Pca recevra la première adresse du réseau et Pcb la deuxième en plus de pouvoir échanger des fichiers à SF de manière sécurisé et de pouvoir être joins depuis n'importe qu'elle machine du réseau, R_P lui prendra la dernière adresse de la plage.

Dans la dernière zone, la zone client ; les machines Pcd et Pcc devront recevoir leur adresse de manière dynamique par l'intermédiaire de R_C qui lui aura une configuration pérenne et sera aussi le routeur qui servira de serveur DHCP pour les machines du réseau.

Analyse du problème et de l'architecture présentée

Analyse théorique de l'exercice :

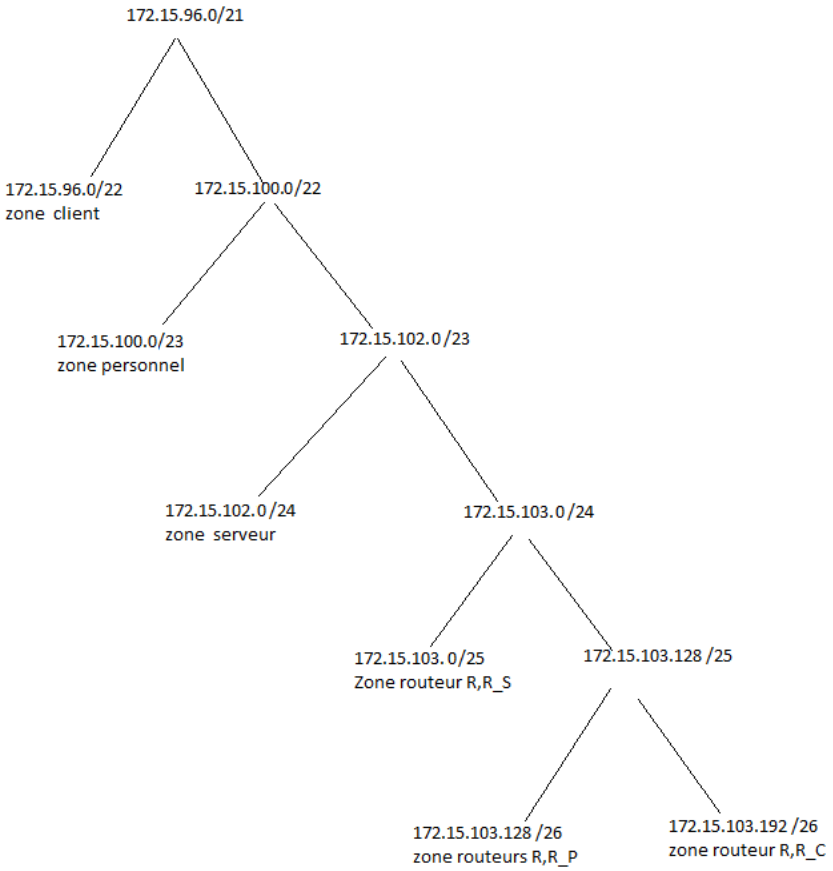
Pour ce problème on avait le sujet 4, ou la valeur de m_1 était 289, celle de m_2 était 700 et celle de m_3 était de 450. Pour ce qui est de la plage d'adresse celle donné était 172.15.96.0/21. Pour répartir les adresses réseaux j'ai utilisé un schéma pour rediviser l'adresse de base en passant de d'un CIDR de /21 à /22 jusqu'à /26.

Tableau d'adressage :

id	Machine min	Plage DHCP nécessaire	Max de machine dispo
Zone personnel	289	De 172.15.100.1/23 à 172.15.101.254/23	510
Zone client	700	De 172.15.96.1/22 à 172.15.99.254/22	1022
Zone serveur	X	De 172.15.102.1/24 à 172.15.102.254/24	254
Zone routeur	6	De 172.15.103.1/24 à 172.15.103.254/24	254
Zone routeur R/R_S	2	De 172.15.103.1/25 à 172.15.103.126/25	126
Zone routeur R/R_P	2	De 172.15.103.129/26 à 172.15.103.190/26	62
Zone routeur R/R_C	2	De 172.15.103.193/26 à 172.15.103.254/26	62

Schéma de séparation :

Le schéma des divisions peut être représenté comme ci-dessous :



En faisant une décomposition telle quelle on arrive à mettre le bon nombre de machines dans chaque réseau.

Création du lab.conf :

Le fichier lab.conf est comme ci-joint:

```
r[bridged]=true
r[0]=net0
rs[0]=net0
r[1]=net1
rp[0]=net1
r[2]=net2
rc[0]=net2
rs[1]=net3
sf[0]=net3
rp[1]=net4
pca[0]=net4
pcb[0]=net4
rc[1]=net5
pcc[0]=net5
pcd[0]=net5
```

Pour ce qui est de l'affectation des adresses IP :
SF : 172.15.102.1, la première adresse du réseau de la zone serveurs.
R_S : 172.15.102.254 sur eth1 qui sera dans la zone serveurs et 172.15.103.2 sur eth0 qui sera connecté à R.
PCA : 172.15.100.1, la première adresse du réseau de la zone personnel.

PCB : 172.15.100.2, la deuxième adresse du réseau de la zone personnel.
R_P : 172.15.101.254 sur eth1 la dernière adresse du réseau de la zone personnel
et 172.15.103.130 sur eth0 qui sera connecté à R.
PCC et PCD : qui reçoivent leur adresse de manière dynamique via le serveurs DHCP sur R_C
R_C : 172.15.99.254 sur eth1 qui sera dans la zone clients
et 172.15.103.194 sur eth0 qui sera connecté à R.

Pour finir il y aura la machine R :
sur eth0 : 172.15.103.1 qui sera connecté à R_S
sur eth1 : 172.15.103.129 qui sera connecté à R_P
sur eth2 : 172.15.103.193 qui sera connecté à R_C
sur eth3 : qui sera le BRIDGED et permettra d'accéder à internet

Configuration précise sur chaque machines

Exemple de configuration classique :

Dans cette partie on expliquera nos choix et notre logique derrière chaque configuration de chaque machine.
Pour ce qui est de la logique global, on a utilisé les fichier **interfaces**, le **/shared** ainsi que les fichiers **.startup** qui nous permettrons de récupérer les informations directement depuis un fichier partagé.

Pour ce qui est des machines SF, PCA et PCB la configuration est quasiment la même.
Dans un premier temps on crée le fichier interfaces de la machines de la manière suivante :

La création de l'interface de loopback, puis l'affectation de l'adresse IP sur eth0 ainsi que la route à prendre lors d'un échange avec une autre machine via le « gateway ».

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 172.15.100.1
netmask 255.255.254.0
gateway 172.15.101.254
```

Ensuite on utilise le .startup pour déplacer ce fichier au bonne endroit, on démarre le protocole de la configuration pérenne et on ajoute le DNS dans le **resolv.conf** à l'aide d'un « echo » et d'une redirection.

```
cp /shared/PCAinterfaces /etc/network/interfaces
/etc/init.d/networking restart
echo "nameserver 8.8.8.8"> /etc/resolv.conf
```

Configuration des routeurs :

Pour ce qui est des routeurs R_P et R_S on utilise aussi une configuration pérenne mais sur les quelles on va aussi précisé les routes à prendre pour chaque fichier.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 172.15.103.130
netmask 255.255.255.192
gateway 172.15.103.129

auto eth1
iface eth1 inet static
address 172.15.101.254
netmask 255.255.254.0
```

Ainsi que le startup qui est exactement le même que précédemment :

```
cp /shared/RPinterfaces /etc/network/interfaces
/etc/init.d/networking restart
echo "nameserver 8.8.8.8"> /etc/resolv.conf
```

Configuration de machine sur DHCP puis du serveur DHCP:

De leur côte, pour PCC et PCD comme il était demandé que leur adresse soit donné de manière dynamique via DHCP on change légèrement leur fichier interfaces pour spécifié un adressage dynamique :

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

On changera aussi le .startup pour lui ajouter le fichier de configuration du DHCP (ifcfg-eth0) ainsi que l'ordre « dhclient » pour démarrer le DHCP sur la machine.

Fichier ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

fichier .startup

```
cp /shared/PCCinterfaces /etc/network/interfaces
/etc/init.d/networking restart
cp /shared/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0
dhclient
```

Ensuite vient le routeur/serveur R_C qui lui va gérer le DHCP, la configuration du fichier dhcpd.conf est la suivante :

```
ddns-update-style none;
subnet 172.15.96.0 netmask 255.255.252.0{
    range 172.15.96.1 172.15.98.188 ;
    default-lease-time 21600 ;
    max-lease-time 43200 ;
    option routers 172.15.99.254 ;
    option domain-name-servers 8.8.8.8;
}
```

et le fichier isc-dhcp-server lui sera comme ça :

```
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
DHCPDv4_PID=/var/run/dhcpd.pid
INTERFACESv4="eth1"
```

Pour ce qui est de la configuration du service DHCP, il nous était demandé qu'au moins 700 machines puissent recevoir leur adresse de manière dynamique.
Dans un premier temps on définit le subnet qui est l'adresse du réseau (ici 172.15.96.0) ainsi que son CIDR sous la forme de netmask (ici /22 = 255.255.252.0), ensuite on spécifie la plage d'adresse qui sera adressé dynamiquement la « range », ensuite on met les paramètres classique : le temps d'adressage, son maximum, son routeur (ici c'est l'adresse de R_C) et le serveurs DNS (ici c'est celui de google).
Il nous manques donc le fichier .startup de R_C et son fichier « interfaces » :

```
cp /shared/RCinterfaces /etc/network/interfaces
/etc/init.d/networking restart
echo "nameserver 8.8.8.8" > /etc/resolv.conf
apt update
apt install -y isc-dhcp-server

cp /shared/dhcpd.conf /etc/dhcp/dhcpd.conf
cp /shared/isc-dhcp-server /etc/default/isc-dhcp-server
/etc/init.d/isc-dhcp-server start
```

Dans la même logique que pour les autres machines, on déplacer les fichiers depuis le /shared vers le bonne endroit. Ensuite on va installer le serveur DHCP en y ajoutant le paramètre -y qui permet d'accepter automatiquement tous ce qui peux nous être demandé pendant l'installation du service, on fini donc par le déplacement des fichier dhcpd.conf et isc-dhcp-server puis le démarrage du serveur dhcp.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 172.15.103.194
netmask 255.255.255.192
gateway 172.15.103.193

auto eth1
iface eth1 inet static
address 172.15.99.254
netmask 255.255.252.0
```

Pour ce qui est du fichier interfaces il ressemblera à la même chose que les autres routeurs :

Configuration du routeur BRIDGED :

La dernière machine à configurer est R, R est très important dans notre architecture car il sera la passerelle entre le réseau interne et internet en plus d'être la passerelle entre toute les zones du réseau interne, il aura donc un fichier interfaces très complet ainsi que quelque commande en plus dans son fichier .startup.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 172.15.103.1
netmask 255.255.255.128
post-up ip route add 172.15.102.0/24 via 172.15.103.2 dev eth0

auto eth1
iface eth1 inet static
address 172.15.103.129
netmask 255.255.255.192
post-up ip route add 172.15.100.0/23 via 172.15.103.130 dev eth1

auto eth2
iface eth2 inet static
address 172.15.103.193
netmask 255.255.255.192
post-up ip route add 172.15.96.0/22 via 172.15.103.194 dev eth2
```

Les routes sur R seront le plus important pour le bon fonctionnement du réseau, ce qu'on fait est donc simple, pour chaque réseau destination on lui donne la prochaine sortie à prendre.

Pour ce qui est du fichier r.startup, on pense à lui ajouter la commande MASQUERADE pour que les autres machines puissent accéder à internet (sur kathara le bridged se fait sur la dernière adresse de la machine, ici c'est donc eth3).

```
cp /shared/Rinterfaces /etc/network/interfaces
/etc/init.d/networking restart
echo "nameserver 8.8.8.8" > /etc/resolv.conf
iptables -t nat -A POSTROUTING -o eth3 -j MASQUERADE
```

Configuration du service d'accès à distance et du service de transfert de fichiers :

Pour finir on nous indiqué qu'un utilisateurs de la zone personnel est souvent avec les clients, cet utilisateurs aimerait donc pouvoir se connecter à pcb depuis la zone client. De plus cette personne

interagit souvent avec la machine SF situé dans la zone serveurs, elle souhaite donc pouvoir retirer et déposer des fichiers sur SF, bien sur tous cela devra être fait de manière sécurisé.

Pour faire ça on utilisera deux choses :
- le protocole ssh
- et le protocole ftp (ici sftp)
(Ces deux protocoles seront détaillé plus loin dans le rapport).

Dans un premier tant on va créer le canal sécurisé sur toute les machines (le protocole ssh), ensuite on va créer notre utilisateurs ainsi que son mot de passe puis on fini par installer le protocole ftp. Ce qui donnera ça dans pcb.startup :

```
cp /shared/PCBinterfaces /etc/network/interfaces
/etc/init.d/networking restart
echo "nameserver 8.8.8.8"> /etc/resolv.conf
systemctl start sshd
useradd -m admin
echo "admin:mdp" | chpasswd
apt update
apt -y install ftp
```

Ici la partie importante pour ssh et ftp commence à la 4 ème lignes.

Du côté du serveur SF on installera le serveur vsftpd qui utilise le protocole ftp mais qui est sécurisé (il passe par ssh c'est pourquoi on doit l'installer avant), on créera la aussi un utilisateur et son mot de passe puis on lui ajoutera la configuration du serveur avant de le démarré, ce qui donne ceci dans sf.startup :

```
cp /shared/SFinterfaces /etc/network/interfaces
/etc/init.d/networking restart
echo "nameserver 8.8.8.8"> /etc/resolv.conf
apt update
apt install -y vsftpd
useradd -m sfadmin
echo "sfadmin:mdp" | chpasswd
cp /shared/vsftpd.conf /etc/vsftpd.conf
/etc/init.d/vsftpd start
```

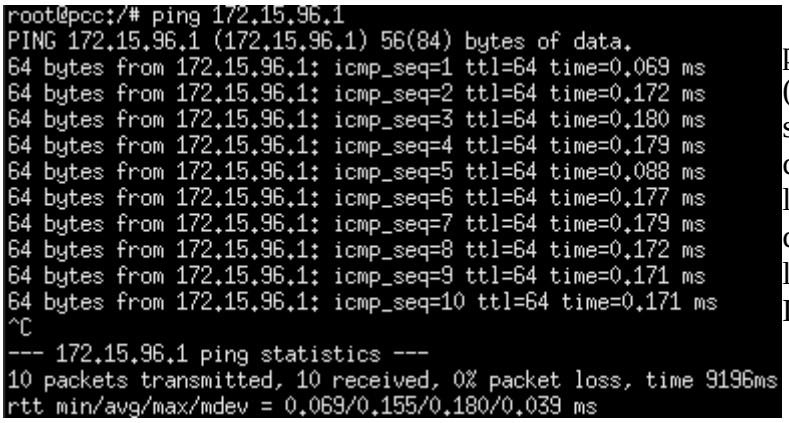
Ensuite le fichier vsftpd.conf sera comme suit :

```
listen=YES
listen_ipv6=NO
anonymous_enable=NO
local_enable=YES
write_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO
```

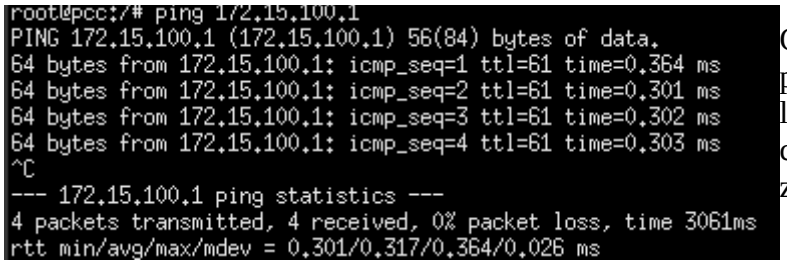
- On le met en écoute
- on désactive l'ipv6 car on est en ipv4
- on désactive les connections anonymes
- on précise que l'on doit être dans le réseau local pour s'y connecter
- que l'on puisse écrire dedans (pour déposer des fichiers et les récupérer par exemples
- la connexion se fait par le port 20.

Différente capture de test ainsi que leur explication :

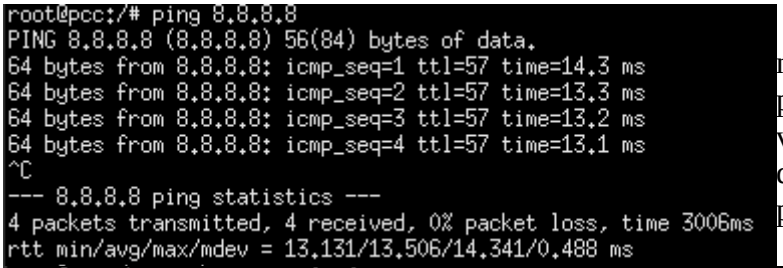
Voici ensuite les résultats de certains test permettant de témoigner du bon fonctionnement de la configuration énoncé ci-dessus.



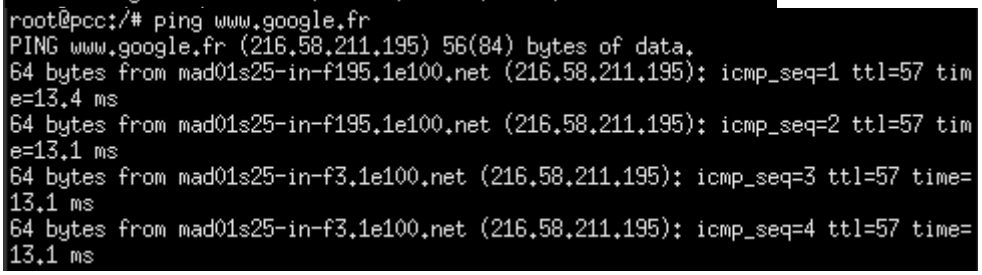
Ici c'est un ping depuis la machine pcc (172.15.96.X) vers la machine pcd (172.15.96.1), on voit qu'il abouti sans problème, on peut même constater que pcc et pcd on bien reçu leur adresse de manière dynamique car elles peuvent communiquer et que leur adresse est dans la plage du DHCP.



Cette capture est un ping entre pcc et pca(172.15.100.1), on voit donc que les machines d'une zone peuvent communiquer avec celle d'une autre zone.



Ce ping permet de monter que n'importe qu'elle machine (ici c'est pcc) peut communiquer avec internet via adresse et le suivant montre que le dns fonctionne correctement car on peut le faire par nom de domaine.



Ici on tente de se connecter depuis une machine de la zone client (ici la machine à la machine à l'adresse 172.15.96.3) sur pcb, pour se faire on rentre la commande ssh suivit du nom de l'utilisateur (ici admin) puis de l'adresse de la machine sur la quelle on veut se connecter (ici pcb),

ensuite le mot de passe nous est demandé puis on est bien connecté sur pcb comme nous le confirme la ligne 3.

```
root@pcb:/# ftp -p 172.15.102.1
Connected to 172.15.102.1.
220 (vsFTPd 3.0.3)
Name (172.15.102.1:root): sfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Ensuite depuis pcb on se connecte au ftp de SF avec la commande ftp -p 172.15.102.1 (le -p permet de se connecter en mode passif et d'éviter les problèmes lié à un potentiel par-feu)

Après s'être connecté au ftp on va :

obtenir un fichier nommé sftest depuis sf

```
ftp> get sftest
local: sftest remote: sftest
229 Entering Extended Passive Mode (|||22709|)
150 Opening BINARY mode data connection for sftest (0 bytes).
     0        0.00 KiB/s
226 Transfer complete.
```

déplacer un fichier nommé pcbtest sur sf

```
ftp> put pcbtest
local: pcbtest remote: pcbtest
229 Entering Extended Passive Mode (|||7481|)
150 Ok to send data.
     0        0.00 KiB/s
226 Transfer complete.
```

On voit d'après l'instruction « Transfer complete » que le déplacement à bien fonctionné.
Voici qui conclu la phase de test.

Explication ftp, shh et capture Wireshark

Explication ftp :

FTP est un protocole standard qui s'établit à la couche application du modèle OSI, utilisé pour le transfert de fichiers entre un client et un serveur (ici entre pcb et SF), ici on a utilisé le service sftp (ssh ftp) qui est plus sécurisé.

Explication ssh :

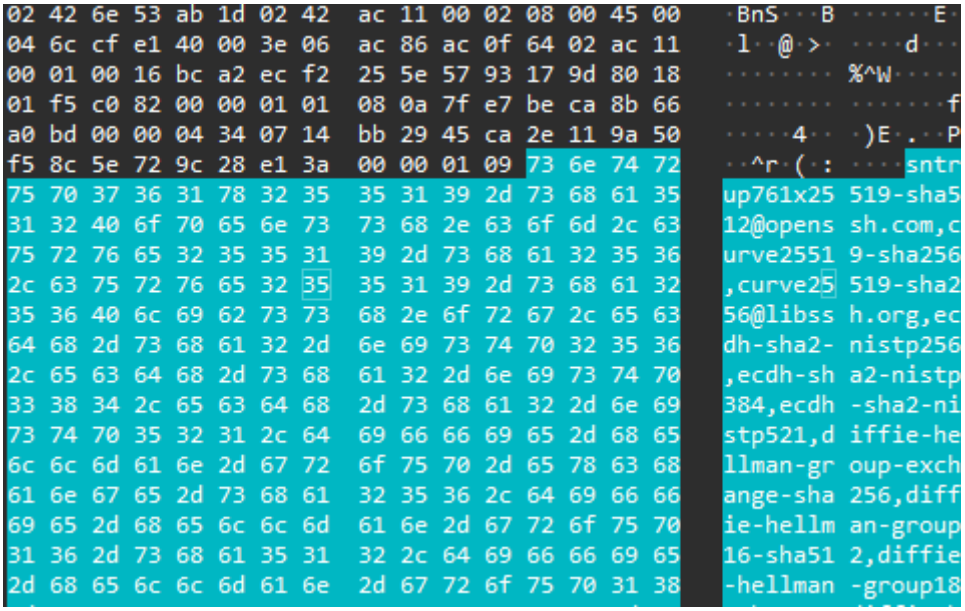
SSH lui est un service permettant :
- d'assurer une connexion sécurisé à distance
- de faire du transfert de fichier de manière sécurisé (sftp vu précédemment)

Tous ça de manière crypté.
Pour ce faire on doit initialisé une connexion avec notamment la commande ssh suivit de l'adresse de destination, ensuite il va y avoir un échange de clef permettant la connexion de manière sécurisé, pour finir l'utilisateur va se connecter à l'aide d'un identifiants et d'un mot de passe, il pourra ensuite bénéficier de tout se que la machine sur la quelle il s'est connecté a à offrir.

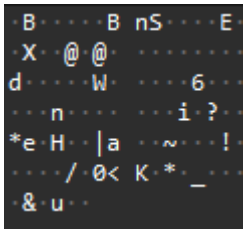
Explication Wireshark :

1	0.000000000	172.17.0.1	172.15.100.2	TCP	74 48290 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
2	0.000127739	172.15.100.2	172.17.0.1	TCP	74 22 → 48290 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=146
3	0.000167571	172.17.0.1	172.15.100.2	TCP	66 48290 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=233875
4	0.000293418	172.17.0.1	172.15.100.2	SSHv2	106 Client: Protocol (SSH-2.0-OpenSSH_8.4p1 Debian-5+deb11u3)
5	0.000305820	172.15.100.2	172.17.0.1	TCP	66 22 → 48290 [ACK] Seq=1 Ack=41 Win=65152 Len=0 TSval=21458
6	0.009042707	172.15.100.2	172.17.0.1	SSHv2	98 Server: Protocol (SSH-2.0-OpenSSH_9.2p1 Debian-2)
7	0.009047539	172.17.0.1	172.15.100.2	TCP	66 48290 → 22 [ACK] Seq=41 Ack=33 Win=64256 Len=0 TSval=2338
8	0.009132214	172.17.0.1	172.15.100.2	SSHv2	1602 Client: Key Exchange Init
9	0.009925533	172.15.100.2	172.17.0.1	SSHv2	1146 Server: Key Exchange Init

On voit sur cette capture wireshark entre pcb et sf que le protocle ssh s'active à la ligne 1 et 2 et on note surtout qu'à la ligne 8 et 9, on a « l'échange de clef » qui va assurer la sécurité des prochaines communications, on peut notamment le voir dans le contenue de ces messages qui sont complètements crypté :



On voit ensuite que sur tous les messages suivant de la capture qu'aucun message ne sera lisible car ils seront tous crypté.



Conclusion

Dans cette SAE d'installation de services réseau on aura donc vu plusieurs choses :
Comment séparer la plage d'adresse donné en zone :
- En augmentant le CIDR pour créer deux parties distincts

Comment configurer des adresses ip et des routes pour pouvoir communiquer avec toutes les machines du réseau :
- À l'aide de fichier interfaces et de fichier startup

Comment faire pour que toutes les machines du réseau puissent accéder à internet :
- En utilisant un BRIDGED sur le routeur principal et en lui donnant un MASQUERADE

Comment faire de l'adresse statique et dynamique par serveur DHCP :
- En installant le protocole et service sur la bonne machine puis en spécifiant un adressage dhcp dans les fichiers interfaces des machines concerné

Comment créer un canal sécurisé de transition de fichier via ssh et sftp :

- En créant un serveur ssh puis en l’utilisant pour avoir un serveur SFTP crypté et donc sécurisé.

Cette SAE m’a pris beaucoup de temps notamment parce que je me suis retrouvé seul, mais je l’es trouvé très intéressante et amusante.