Prérequis :

Sur la machine hôte, repérer le nom de l'interface Ethernet qui n'est pas utilisée pour la connexion Internet :
- si la connexion Internet s'effectue en wifi , l'interface à utiliser est l'interface Ethernet filaire
- si la connexion Internet s'effectue en filaire, l'interface à utiliser sera une 2nde interface filaire. Si la machine ne possède qu'une interface filaire, utiliser un adapteur de type USB-Ethernet.

Sur la machine virtuelle SR07-GNS3, configurer 2 interfaces réseau :

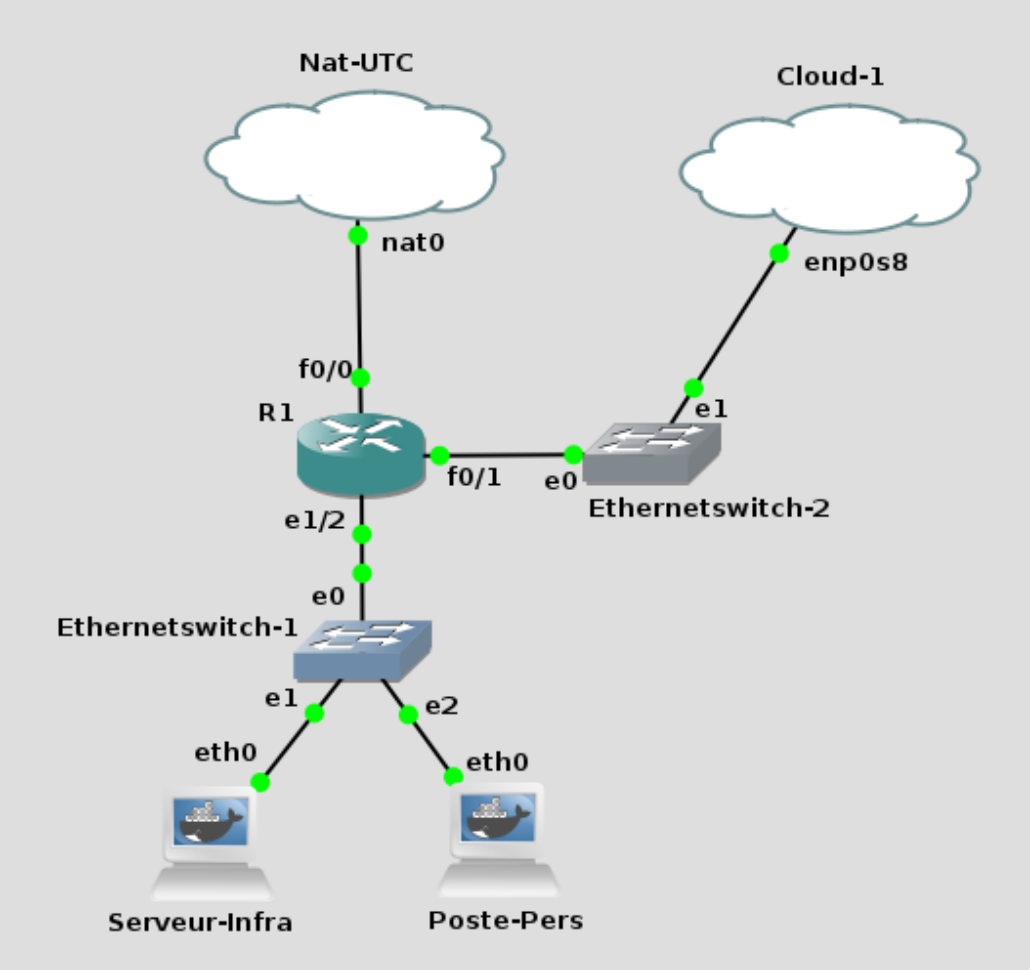
1ère interface :
- Type NAT
- Câble branché

2nde interface :
- Type Bridge (accès par pont)
- Nom = celui repéré à l'étape précédente (ex : eth0, eth1…)
- Mode promiscuité : Tout autoriser
- Câble branché

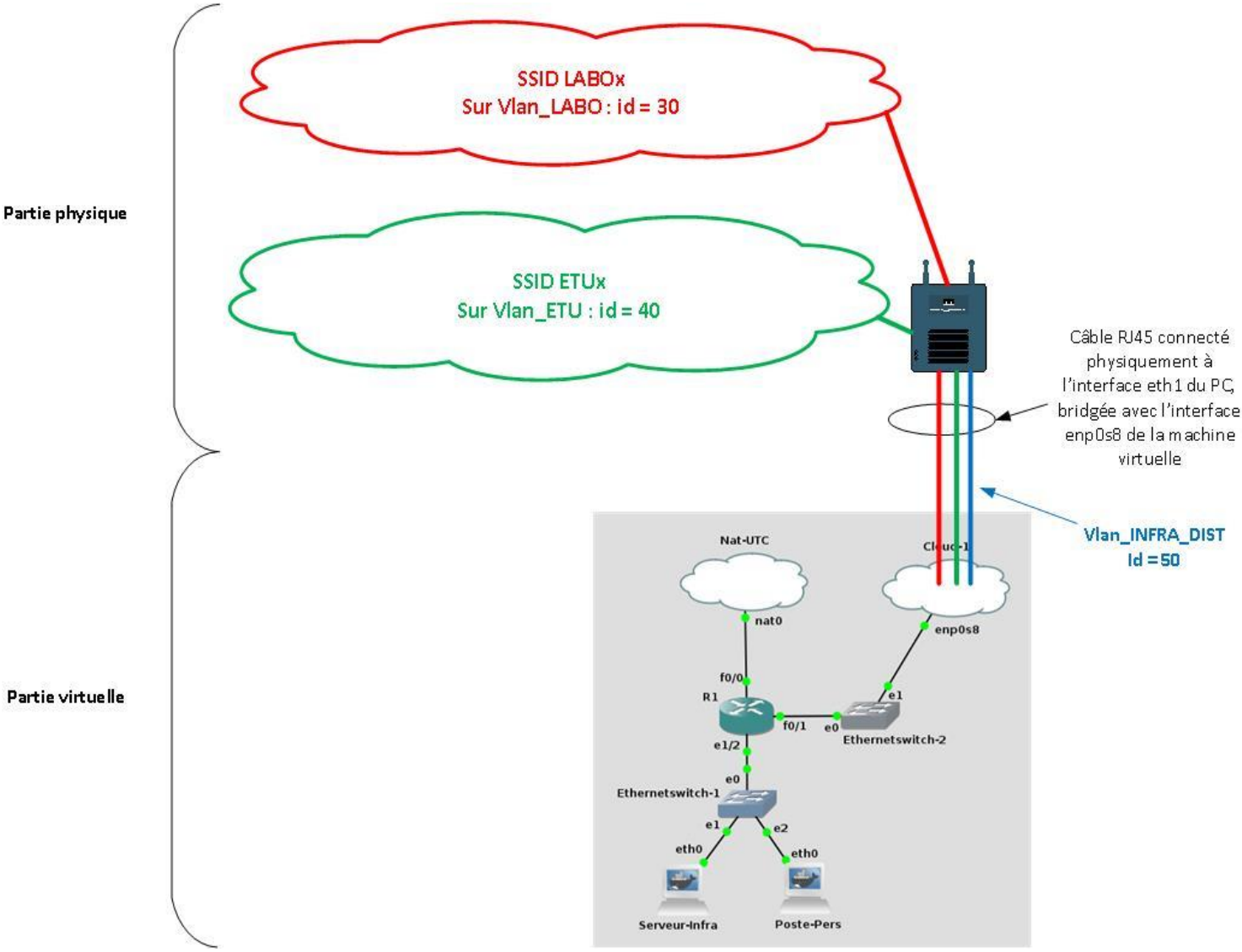
Le TD 4 consiste à donner suite au TD-A1 « Architectures Résilientes : technologies des réseaux locaux sécurisés ».

Le TD 4 peut s'effectuer juste après l'exercice 5 (DHCP) du TD1.

Architecture cible virtualisée :



Architecture cible complète :



1. Configuration des interfaces vlan sur le routeur :

Configurer le routeur de manière à ce qu'il desserve 3 vlans sur l'interface FastEthernet0/1 :

- vlan 30 :
  - nom= **LABO**
  - ip address 10.0.30.1 255.255.255.0
  - ip nat inside
- vlan 40 :
  - nom= **ETU**
  - ip address 10.0.40.1 255.255.255.0
  - ip nat inside
- vlan 50 :
  - nom= **INFRA_DIST**
  - ip address 10.0.50.1 255.255.255.0
  - ip nat inside

2. Configuration des pools DHCP pour chaque VLAN

Sur le routeur, configurer un pool DHCP afin qu'il puisse délivrer des adresses IP sur les vlans LABO, ETU et INFRA_DIST.

3. Configuration d'un switch pour la desserte des vlans

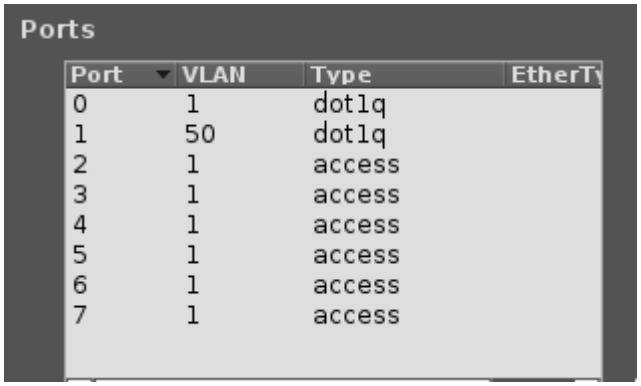
Connecter un switch de l'interface Ethernet0 sur l'interface FastEthernet0/1 du routeur. L'interface Ethernet 0 du switch sera de type dot1q avec un vlan natif = 1

4. Mapping de l'infrastructure virtuelle à une interface physique

Connecter un Cloud mappé sur l'interface **enp0s8** (soit la 2nde interface de la machine virtuelle correspondante à l'interface de la machine hôte repérée dans les prérequis) et le connecter sur l'interface Ethernet1 du switch.

L'interface Ethernet1 sera de type dot1q avec le vlan 50 comme vlan natif. Ceci signifie que tous les paquets arrivant sur l'interface e1 seront forwardés dans le vlan 50.

-



5. Configuration de la borne wifi par l'interface Console

Se connecter à l'interface console de la borne wifi en utilisant un câble série RS232 de type DB9-RJ45. L'interface DB9 sera connectée soit directement sur le PC soit en utilisant un adapteur USB-DB9.

Si un adaptateur SUB est utilisé, vérifier sa détection avec la commande **dmesg** :

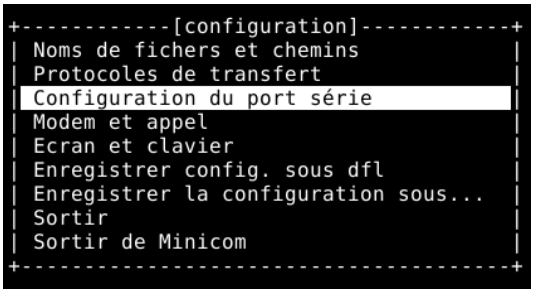
```
# dmesg
```

…
[828746.163406] usb 1-3: FTDI USB Serial Device converter now attached to **ttyUSB0**

Sur la machine hôte, lancer l'utilitaire minicom en mode setup :

```
# minicom -s
```

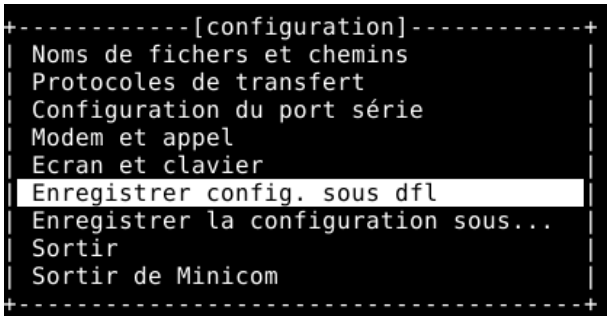
Sélectionner le menu Configuration du port série :



Configurer le port série avec les informations suivantes :

device : **/dev/ttyS0** pour une machine avec un port DB9
device : **/dev/ttyUSB0** pour une machine avec un adapteur USB-DB9
vitesse : **9600 bit/s**
bits data : **8**
parité : **N**
bits de stop : **1**
contrôle de flux matériel : **N**
contrôle de flux logiciel : **N**

Enregistrer la configuration en tant que config par défaut :



Sortir de minicom.

Relancer l'utilitaire minicom sans l'option -s .

```
# minicom
```

Appuyer plusieurs fois sur Entrée et le prompt **ap>** doit apparaître .

```
Bienvenue avec minicom 2.7

OPTIONS: I18n
Compilé le Apr 25 2017, 21:09:25.
Port /dev/ttyUSB0, 15:51:31

Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales


ap>
```

6. Connexion de la borne wifi à l'infrastructure virtuelle GNS3

Connecter un câble Ethernet entre la borne wifi sur l'interface Ethernet disponible sur le PC (à défaut sur l'adapteur USB-RJ45).

Vérifier que l'interface de la borne wifi récupère une adresse IP délivrée par le routeur.

Entrer en mode **enable** (le mot de passe par défaut est **Cisco**)

```
ap>enable
Password:        → Cisco
ap#show running-config interface BVI1
Building configuration...

Current configuration : 82 bytes
!
interface BVI1
 ip address dhcp client-id FastEthernet0
 no ip route-cache
end

ap#sh ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
BVI1                   10.0.50.2       YES DHCP   up                    up
Dot11Radio0            unassigned      YES unset  administratively down down
Dot11Radio1            unassigned      YES unset  administratively down down
FastEthernet0          unassigned      YES other  up                    up
ap#
```

Le cas échéant , relancer la requête DHCP :

```
ap#conf t
ap(config)#int BVI1
ap(config-if)#shutdown
```

```
ap(config-if)#no shutdown
*Mar  1 22:49:34.462: %LINK-5-CHANGED: Interface BVI1, changed state to
administratively down
*Mar  1 22:49:36.752: %LINK-3-UPDOWN: Interface BVI1, changed state to up
*Mar  1 22:49:40.630: %DHCP-6-ADDRESS_ASSIGN: Interface BVI1 assigned
DHCP address 10.0.50.4, mask 255.255.255.0, hostname ap
ap(config-if)#end
```

7. Vérifier la connexion Internet

```
ap#ping 10.0.50.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/209/1007 ms
```

```
ap#ping 195.83.155.55
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/23/39 ms
```

```
ap#ping www.utc.fr
Translating "www.utc.fr"...domain server (195.83.155.55) [OK]
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/20/21 ms
```

8. Récupération des fichiers de configuration sur Serveur_Infra

Sur le routeur , désactiver les access-lists potentielles sur le vlan INFRA (id=10) :

```
R# conf t
R(config)# interface e1/2.10
R(config-if)# no ip access-group INFRA_IN in
R(config-if)# no ip access-group INFRA_OUT out
R(config-if)# end
R# show run int e1/2.10
```

Sur Serveur_Infra :

```
# usermod -d /etc/sr07 -m -s /bin/bash sr07
#  su - sr07
$ wget http://www.utc.fr/~quetwilf/sr07/wifi_vlan_infra.txt
$ wget http://www.utc.fr/~quetwilf/sr07/wifi_vlan_labo.txt
$ wget http://www.utc.fr/~quetwilf/sr07/wifi_vlan_etu.txt
$ wget http://www.utc.fr/~quetwilf/sr07/wifi_ssid_labo.txt
$ wget http://www.utc.fr/~quetwilf/sr07/wifi_ssid_etu.txt
$ wget http://www.utc.fr/~quetwilf/sr07/wifi_ssid_etu_mac.txt
```

```
$ wget http://www.utc.fr/~quetwilf/sr07/wifi_radius.txt
```

```
$ exit
# echo “KexAlgorithms   diffie-hellman-group1-sha1” >> /etc/ssh/sshd_config
# echo “Ciphers  aes256-cbc”  >> /etc/ssh/sshd_config
# /etc/init.d/ssh restart
```

9. Configuration des vlans sur la borne wifi

Configuration du vlan INFRA_DIST sur la borne wifi

```
ap#copy scp://sr07:sr07sr07@10.0.10.3/wifi_vlan_infra.txt run
Destination filename [running-config]?
 Sending file modes: C0644 749 wifi_vlan_infra.txt
!
749 bytes copied in 0.561 secs (1335 bytes/sec)
```

Configuration du vlan LABO sur la borne wifi

```
ap#copy scp://sr07:sr07sr07@10.0.10.3/wifi_vlan_labo.txt run
Destination filename [running-config]?
 Sending file modes: C0644 749 wifi_vlan_labo.txt
!
749 bytes copied in 0.561 secs (1335 bytes/sec)
```

Configuration du vlan ETU sur la borne wifi

```
ap#copy scp://sr07:sr07sr07@10.0.10.3/wifi_vlan_etu.txt run
Destination filename [running-config]?
 Sending file modes: C0644 749 wifi_vlan_etu.txt
!
749 bytes copied in 0.561 secs (1335 bytes/sec)
```

10. Configurer les SSID sur le vlan LABO

Le vlan LABO sera configure avec une sécurité de type **WPA2-PSK** (clé pré partagée) avec cryptage **AES-256**.

Sur le compte **sr07** de la machine virtuelle, éditer le fichier **wifi_ssid_labo.txt** de manière à remplacer le nom du SSID génériques au format laboXX et etuXX par des SSIDs portant le noms de votre groupe. Ceci permettra d'éviter les doublons au sein de la salle.

Tester la connexion sur le SSID etuXX.

11. Configurer le SSID sur le vlan ETU

Le vlan ETU aura une sécurité de type **WPA2 Enterprise** avec cryptage **AES-256**.

La configuration de la sécurité WPA2 Enterprise nécessite la mise en place d'un serveur d'Authentification Radius. Ce serveur sera installé sur Serveur_Infra au point 12 .

Valider le contenu du fichier **wifi_radius.txt**.

Dans cet exemple , le serveur radius permettra l'authentification du client possédant l'adresse 10.0.10.3.

```
ap#copy scp://sr07:sr07sr07@10.0.10.3/wifi_radius.txt run
Destination filename [running-config]?
 Sending file modes: C0644 749 wifi_radius.txt
!
749 bytes copied in 0.561 secs (1335 bytes/sec)
```

Sur le compte **sr07** de la machine virtuelle, éditer le fichier **wifi_ssid_etu.txt** de manière à remplacer le nom du SSID **etuXX** par un SSID portant le noms de votre groupe. Ceci permettra d'éviter les doublons au sein de la salle.

```
ap#copy scp://sr07:sr07sr07@10.0.10.3/wifi_ssid_etu.txt run
Destination filename [running-config]?
 Sending file modes: C0644 749 wifi_ssid_etu.txt
!
749 bytes copied in 0.561 secs (1335 bytes/sec)
```

12. Configurer l'authenfication EAP sur Serveur_Infra.

L'installation du package **Freeradius** permettra de fournir les services d'authentification.

Configurer le fichier **/etc/freeradius/3.0/clients.conf** :

```
clients ap-groupxx {                    ← avec xx= numéro de groupe
        ipaddr = 10.0.50.2              ← avec l'adresse IP récupérée par la borne
        secret = sr07sr07
}
```

Configurer le fichier **/etc/freeradius/3.0/users** :

```
etu1    Cleartext-Password:= « etu1 »
        Service-Type = Framed-User
```

Dans le fichier **/etc/freeradius/3.0/radiusd.conf** , remplacer **auth = no** par **auth = yes**

Le module eap est chargé dans la section authorize {} du fichier **/etc/freeradius/3.0/sites-enabled/default**

Le module eap est configuré dans le fichier **/etc/freeradius/3.0/mods_enable/eap**

Dans ce fichier, modifier le **default-eap-type** à **peap**.

Arrêter le service freeradius :

```
# /etc/init.d/freeradius stop
```

Lancer le service freeradius en mode debug :

```
# /usr/sbin/freeradius  -X
```

Si aucun message n'arrive sur la console de freeradius , il faut valider que l'échange s'effectue correctement en lançant une capture sur Serveur_Infra :

```
# tcpdump -n port 1812
```

Le client se connectant au réseau etuXX doit recevoir un message d'alerte spécifiant que le certificat est non valide. Effectuer le test de connexion depuis un smartphone par exemple.

13. Configurer l'authentification par adresse MAC

Repérer l'adresse mac du client qui effectue la connexion wifi (ex : aa:11:bb:22:cc:33).

Sur le serveur radius, ajouter une entrée au fichiers **/etc/freeradius/3.0/users** en respectant le format de l'adresse mac ci-dessous :

```
aa11bb22cc33          Cleartext-Password:= « aa11bb22cc33 »
                      Service-Type = Framed-User
```

Redémarrer le serveur Radius en mode debug.

Editer le fichier **wifi_ssi_etu_mac.txt** et modifier le nom du SSID etuXX par le nom du SSID correspondant au groupe.

Charger le fichier sur a borne wifi :

```
ap#copy scp://sr07:sr07sr07@10.0.10.3/wifi_ssid_etu_mac.txt run
Destination filename [running-config]?
 Sending file modes: C0644 749 wifi_ssid_etu_mac.txt
!
749 bytes copied in 0.561 secs (1335 bytes/sec)
```

14. Configuration d'une PKI sur le serveur radius

A l'aide d'un PKI, créer un certificat serveur pour le serveur Radius.

Sur le serveur Radius, télécharger les fichiers préalablement créés, à savoir le certificat du serveur, la clé privée, le certificat de l'authorité.

De manière à pouvoir utiliser cette PKI, dans la section tls-config tls-common {}du fichier **/etc/freeradius/3.0/mods_enable/eap**, , modifier les lignes de configuration suivantes :

- private_key_password
- private_password
- certificate_file
- ca_file
- ca_path