COMPTE RENDU SAE12

Pendichev Andon

Exercice 1 : Configuration des Machines

Objectif : Configurer *les adresses IP et activer les interfaces réseau pour permettre la communication entre les machines.*

R (Routeur)

Commandes :

```
ip addr add 10.1.0.254/24 dev eth0 ip
link set eth0 up
ip addr add 10.2.0.254/24 dev eth1 ip
link set eth1 up
```

Explications:

ip addr add 10.1.0.254/24 dev eth0

Cette commande attribue l'adresse IP 10.1.0.254 à l'interface eth0 avec un masque de sous réseau /24 (255.255.255.0). ip link set eth0 up

Cette commande active l'interface eth0, ce qui est nécessaire pour que la communication fonctionne. Les mêmes commandes sont utilisées pour configurer l'interface eth1 dans le réseau 10.2.0.0/24.

Généralités

Un routeur connecte deux réseaux distincts. Ici, il relie 10.1.0.0/24 (eth0) et 10.2.0.0/24 (eth1).
Chaque interface du routeur doit avoir une adresse IP unique.

Exercice 2 : Configuration des interfaces et tables de routage

Pour attribuer une adresse IP :
Exemple avec la machine nfs : on tape dans le terminal de la machine : ip addr add 10.1.0.1/24 dev eth0. Ensuite pour activer eth0, on tape : ip link set eth0 up. Il faut répéter la même opération pour les autres machines en respectant les adresses IP.

Pour ajouter une route (un exemple avec une machine de chaque réseau) :

- nfs : ip route add 10.2.0.0/24 via 10.1.0.254 et ensuite ip route add default via 10.1.0.254
- max : ip route add 10.1.0.0/24 via 10.2.0.1 et ensuite ip route add default via 10.2.0.1 - R : ip route add default via 10.2.0.2

Résultat si à partir de nfs on lance un ping vers pirate (deux réseaux différents) :

```
PING 10.2.0.4 (10.2.0.4) 56(84) bytes of data.
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
```

Même chose mais cette fois ci de pirate vers nfs :

```
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
--- 10.1.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4026ms
```

Pour vérifier si R peut ouvrir une connexion SSH :

- A partir de nfs : on tape la commande : /etc/init.d/ssh start

Résultat : `[ ok ] Starting OpenBSD Secure Shell server: sshd.`

- Sur R : on tape la commande : ssh root@10.1.0.1

Résultat : `ssh: connect to host 10.1.0.1 port 22: No route to host`

Généralités : Une route par défaut est essentielle pour connecter une machine à d'autres réseaux. Le routeur R agit comme une passerelle pour transmettre les paquets entre les réseaux.

Exercice 3 : Test de l'accès au monde extérieur

Pour récupérer la machine physique et l'adresse du serveur DNS sur la machine physique :

- adresse ip : on tape la commande ip link show Résultat : 172.19.12.240/20
- adresse dns : cat /etc/resolv.conf

Résultat : 10.255.255.254

Objectif :

Tester la connectivité entre le réseau virtuel et le monde réel via la passerelle G1.

Machine : max

1.Test du ping vers la machine physique : ping 192.168.1.41 ,Permet de vérifier si max peut communiquer avec la machine physique.

2.Téléchargement d'un fichier avec wget :

wget http://www.iutv.univ-paris13.fr/index.php

Teste si max peut accéder à Internet et télécharger un fichier.

Généralités : ping est utilisé pour tester la connectivité. wget permet de télécharger des fichiers depuis un serveur HTTP.

Exercice 4 : Sécurisation du routeur R

Sur R, nous devons modifier les paquets INPUT et OUTPUT. Par défaut, on choisit la politique : DROP

Pour configurer les politiques par défaut :

iptables -P INPUT DROP

iptables -P OUTPUT DROP

Nous devons autoriser les paquets ICMP à entrer dans la chaîne INPUT et sortir de la chaîne OUTPUT. Voici les deux commandes :

iptables -A INPUT -p icmp -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT

A partir du routeur, les paquets doivent être autorisés par défaut dans la chaîne FORWARD.

Voici la commande : iptables -
P FORWARD ACCEPT

Depuis nfs, nous allons envoyer un ping au routeur pour vérifier si ce dernier reçoit les pings : ping 10.1.0.254 Résultat :

```
PING 10.1.0.254 (10.1.0.254) 56(84) bytes of data.
--- 10.1.0.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss
```

Le routeur reçoit encore les pings

Nous allons vérifier si R peut envoyer des pings. Envoyons un ping à max : ping 10.1.0.4 Résultat :

```
PING 10.2.0.4 (10.2.0.4) 56(84) bytes of data.
```

Le routeur peut envoyer des pings.

Vérifions si le ssh fonctionne : ssh root@10.1.0.1 . Résultat : `ssh: connect to host`

```
10.1.0.1 port 22: Connection timed out
```

La connexion est coupée. Le ssh ne fonctionne plus.

Exercice 5 : Sécurisation du réseau 10.1.0.0/24

Il faut modifier la chaîne FORWARD. Par défaut, il faut utiliser la politique ACCEPT. Nous allons maintenant permettre uniquement l'accès à nfs depuis le réseau 10.1.0.0/24 : iptables -A FORWARD -d 10.1.0.1/24 -s 10.1.0.0/24 -j ACCEPT iptables -A FORWARD -d 10.1.0.1/24 -j DROP

Il faut maintenant empêcher nfs d'envoyer et de recevoir vers et depuis l'extérieur sauf
10.1.0.0 :
iptables -A FORWARD -s 10.1.0.1 ! -d 10.1.0.0/24 -j DROP
iptables -A FORWARD -d 10.1.0.1 ! -s 10.1.0.0/24 -j DROP

Nous allons bloquer l'accès de pirate au réseau 10.1.0.0 : iptables -A FORWARD -s 10.2.0.3 -d 10.1.0.0/24 -j LOG --log-prefix "BLOCK PIRATE: "
iptables -A FORWARD -s 10.2.0.3 -d 10.1.0.0/24 -j DROP

Depuis pirate, nous allons vérifier si on peut pinger nfs et anna : ping 10.1.0.1 et ping 10.1.0.2
Résultats :

```
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
--- 10.1.0.2 ping statistics ---
```

On ne peut pas pinger nfs et anna.

Depuis nfs, testons un ping vers max : ping 10.2.0.4. Résultat :

```
PING 10.2.0.4 (10.2.0.4) 56(84) bytes of data.
3 packets transmitted, 0 received, 100% packet loss, time
  7039ms
```

C'est un échec.

Depuis max, testons un ping vers nfs : ping 10.1.0.1. Résultat :

```
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
3 packets transmitted, 0 received, 100% packet loss, time
6028ms
```

C'est un échec.

Depuis anna, testons un ping vers max et nfs : ping 10.2.0.4 et ping 10.1.0.1.
Résultats :

```
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
3 packets transmitted, 0 received, 100% packet loss
```

Anna peut pinger nfs mais pas max.

Exercice 6 : Interdiction des paquets SSH

Sur le routeur nous devons bloquer les connexions SSH provenant de machines hors du réseau 10.1.0.0 avec la commande : iptables -A FORWARD -p tcp --dport 22 ! -s 10.1.0.0/24 -j DROP. Nous devons ensuite vérifier que les connexions SSH depuis le réseau 10.1.0.0 fonctionnent avec la commande : iptables -A FORWARD -p tcp --dport 22 -s 10.1.0.0/24 -j ACCEPT.

A partir de pirate, nous allons essayer de se connecter à nfs via le SSH : ssh root@10.1.0.1.
Résultat : `ssh: connect to host 10.1.0.1 port 22: Connection timed out`
Vu que pirate fait partie du réseau 10.2.0.0, la connexion n'est pas possible.

A partir de anna, nous allons essayer de se connecter à nfs via le SSH : ssh root@10.1.0.1.
Résultat :

```
The authenticity of host '10.1.0.1 (10.1.0.1)' can't be established.
ECDSA key fingerprint is e2:16:7e:9a:51:d3:a1:08:9c:be:11:73:de:e6:54:f1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.1.0.1' (ECDSA) to the list of known hosts.
root@10.1.0.1's password:
```

La connexion est réussie.

Exercice 7 : Résolution de nom

Sur chaque machine sauf pirate, il faut modifier le fichier /etc/hosts avec la commande : nano /etc/hosts. Le contenu de chaque fichier sera le même :

10.1.0.1 nfs
10.1.0.2 anna
10.2.0.4 max

Pour vérifier si la résolution des noms fonctionne, nous allons faire des pings. Mais au lieu de mettre l'adresse IP, nous allons mettre le nom des machines où on veut envoyer un ping:

```
ping -c 1 nfs
```

La résolution de nom fonctionne bien car les adresses IP sont reconnues.

Exercice8 : Service de configuration réseau

Objectif :

Configurer un service DHCP sur le routeur R pour attribuer dynamiquement des adresses IP.

Sur R:

Commandes :

1.Installer le service DHCP
apt-get install isc-dhcp-server -y

2.Configurer /etc/dhcp/dhcpd.conf : subnet 10.1.0.0 netmask 255.255.255.0 { range 10.1.0.50 10.1.0.100; option routers 10.1.0.254;
option domain-name-servers 10.1.0.10;}

3.Redamarer le service DHCP:
systemctl restart isc-dhcp-server
4.configurer les clients pour utiliser

DHCP

iface eth0 inet dhcp