RESUME TP

Réseaux

I. Netkit

1. lab.conf

```
LAB_DESCRIPTION="<Description>"
LAB_VERSION=<Version>
LAB_AUTHOR="<Auteur>"
LAB_EMAIL=<Mail>
LAB_WEB=<Adresse web>

<Nom machine>[<Numéro interface>]="<Domaine de collision>"
pc2[mem]= 128 # taille de la RAM de la machine
p2[0]="dca" # exemple
p3[0]="dcb" # exemple
...
```

2. <Nom machine>.startup et dossier <Nom machine>

`<Nom machine>.startup` contient des commandes exécutées au démarrage de la machine.

Le dossier `<Nom machine>` contient des fichiers restaurés à chaque démarrage.

3. Dossier hosthome sur les machines

Sur les machines virtuelles, le dossier `/hosthome` est un alias du dossier home de la machine physique.

II. Commandes usuelles

`ifconfig`	Liste des interfaces réseau
`ifconfig <interf> <up/down>`	Allumer/Eteindre interface réseau
`ifconfig <interf> hw ether <MAC>`	Changer l'adresse MAC
`ifconfig <interf> <IP>`	Changer l'adresse IP
`ping –c [<nbPing>] <ip ou host>`	Envoi d'un ping
`tcpdump –i <interf>`	Récupérer le trafic réseau de la carte
`[-n]`	ne pas faire de RDNS
`[-vvv]`	verbosité
`[-s 0]`	pas de limite de taille de paquet
`[no stp]`	suppr du STP
`[ip]`	uniquement IP
`[-w <nomFichier>]`	enregistrer dans un fichier
`ps –aux`	Liste des services en cours
`netstat –latupen4`	Liste des ports ouverts

III. Démons

`/etc/init.d/<demon> start|stop|restart`	Démarrer/arrêter/redémarrer sur un démon

- **inetd** : telnet, ftp, …
  - /etc/inetd.conf
- **ssh** : serveur SSH
- **dhcp3-server** : serveur DCHP
  - /etc/dhcp3/dhcpd.conf

IV. Commutation

1. Création d'interfaces logiques

Sur `eth0`, il est possible de créer plusieurs interfaces logiques `eth0.0`, `eth0.1`, `eth0.2`, …

2. Table de commutation

`arp`	Afficher la table de commutation
`[-n]`	ne pas faire de rdns
`[-a <hote>]`	pour un hôte
`[-d <hote>]`	supprimer l'entrée

`arpspoof`	Spoofing d'adresse MAC
`[-i <interface>]`	
`[-t <hostFromTarget>]`	
`<hostTo>`	

3. Création de bridge / VLAN

`brctl addbr vlan10`	création du bridge VLAN 10
`brctl addif vlan10 eth0.10`	ajout de eth0.10 au bridge VLAN 10
`brctl addif vlan10 eth1`	ajout de eth1 au bridge VLAN 10
`brctl addif vlan10 ...`	ajout au bridge VLAN 10
`brctl stp vlan10 on`	activation du Spanning Tree Protocol
`ifconfig vlan10 up`	activation du bridge VLAN 10
`brctl showmacs vlan10`	affiche les MAC d'un bridge VLAN 10
`brctl setageing vlan10 <time>`	temps de conservation sur le bridge VLAN 10

V. Routage

1. IP-Aliasing

Pas vraiment du routage, permet de mettre plusieurs IP à une interface.

`ifconfig <interface>:<alias> <nouvelle IP>` crée un IP-aliasing (plusieurs IP sur une interface)

Exemple :

```
ifconfig eth0 10.0.0.1 up
ifconfig eth0:0 20.0.0.1
```

2. Routage UNIX

`sysctl ipv4.forward = 1`	Active le routage UNIX

3. Routage par zebra (Cisco)

a. Démarrage

`/etc/zebra/daemons`	liste des démons et protocoles zebra activés
`/etc/zebra/zebra.conf`	Config zebra, contient le mot de passe (zebra par défaut)
`/etc/init.d/zebra start`	Démarrage zebra
`/etc/init.d/zebra restart`	Redémarrage zebra
`telnet localhost 2601`	Connexion à zebra en local

b. Principe

On peut toujours taper « ? » pour savoir ce qu'il est possible de taper.

Il y a un mode *normal* et un mode *admin*. On change de mode avec `enable`.

c. Exemple de configurations

`Router# configure terminal`	Configure le routeur
`R(config)# interface eth0`	Configure une interface
`R(config-if)# ip address 100.0.0.1/8`	Configure l'IP d'une interface
`R(config)# ip route <IPdest> <IPNextHop>`	Configure un routage
`R(config)# ip route 192.168.2.0/24 100.0.0.2`	Exemple
`R# copy running-config startup-config`	Sauvegarder la config en cours

4. Passerelle par défaut sur un PC

`route add default gw 192.168.1.1`	Ajout de la passerelle par défaut

VI. DNS

1. Introduction

Le serveur DNS d'une zone référence :

- la zone racine
- les zones immédiatement inférieures
- les machines de la zone

2. Fichiers de configuration DNS sur les clients

- `/etc/resolv.conf` : l'IP du serveur DNS à interroger pour les résolutions
  ```
  search <nom zone>
  nameserver <IP DNS>
  ```
- `/etc/hosts` : couple nom/IP statiques
- `/etc/host.conf` : ordre de priorité

3. Commandes pour obtenir enregistrement d'un domaine

`dig <machine>` : affiche les enregistrements DNS

`dig <domaine> <type>` : affiche les enregistrements <type> du domaine

`dig –t ANY <domaine>` : affiche tous les enregistrements du domaine

dig +trace <machine> : affiche la trace de la résolution

`host <machine>` : affiche les enregistrements DNS d'une machine

4. Configuration d'un serveur DNS avec BIND

a. Fichier de configuration générale /etc/bind/named.conf

```
options {
       directory "/var/named";
};

# cas général
zone "<nomZone>" {
        type <master/slave/hint>;
        file "db.<zone>"; # fichier dans /var/cache/bind
};

# toujours indiquer le DNS racine
zone "." {
        type <master/slave/hint>;
        file "db.root";
};

# exemple sur le DNS de la zone fr
zone "fr" {
    type master;
    file "master/houba.maison";
};

# enregistrements pour le localhost et le broadcast
zone "localhost" {
        type master;
        file "db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "db.127";
};
zone "0.in-addr.arpa" {
        type master;
        file "db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "db.255";
};
```

b. Fichier secondaires /var/cache/bind/db.<zone>

i. Cas général

```
@       IN      SOA     <ma.zone>.    <mail>.<ma.zone>. (
                        2013122000 # serial
                        8H   # durée rafraichissement
                        2H   # durée rafraichissement en cas d'échec
                        1W   # durée d'expiration
                        1D ) # durée de vie

# enregistrements DNS
<zone.>             IN    NS       <machine.zone.>   # Nom du DNS d'une zone
<zone.>             IN    MX    5  <machine.zone.>   # Nom du serveur SMTP d'une zone
<machine.zone.>     IN    A        <IPv4>            # IP d'une machine
<machine.zone.>     IN    AAAA     <IPv6>            # IP d'une machine
<machine.zone.>     IN    CNAME    <machine2.zone.>  # Alias de nom pour une machine
```

On peut avoir <zone.> = @ ce qui veut dire « la zone en cours »

<zone.> ou <machine.zone.> peut être soit absolu avec point à la fin, soit relatif sans point à la fin.
Ex : « ns.truc.fr. » ou « ns » dans le fichier de config de la zone truc.fr. sont équivalent.

ii. db.root sur un serveur de la zone « . »

```
@       IN      SOA     ROOT-SERVER. root.ROOT-SERVER. (
                        2005122000
                        28800
                        14400
                        360000
                        0 )

@               IN      NS      ROOT-SERVER.
ROOT-SERVER.    IN      A       192.168.0.5
fr.             IN      NS      nsfr.fr.
nsfr.fr.        IN      A       192.168.0.1
org.            IN      NS      nsorg.org.
nsorg.org.      IN      A       192.168.0.2
```

iii. db.root sur un serveur de la zone « fr », zone « . » en hint

```
.               IN      NS      ROOT-SERVER.
ROOT-SERVER.    IN      A       192.168.0.5
```

iv. db.fr sur un serveur de la zone « fr » en master

```
@       IN      SOA     nsfr.fr.      root.nsfr.fr. (
                        2005121900
                        8H
                        2H
                        1W
                        1D )

@                   IN      NS      nsfr.fr.
nsfr.fr.            IN      A       192.168.0.1
truc.fr.            IN      NS      nstruc.truc.fr.
nstruc.truc.fr.     IN      A       192.168.0.10
```

v. *db.truc sur le serveur de la zone « truc.fr »*

```
@       IN      SOA     nstruc.truc.fr.     root.nstruc.truc.fr. (
                        2005121900
                        8H
                        2H
                        1W
                        1D )

@                       IN      NS      nstruc.truc.fr.
nstruc.truc.fr.         IN      A       192.168.0.10
pc1.truc.fr.            IN      A       192.168.0.11
```

VII. Firewall : filtrage et NAT

```
# regles par défaut
iptables -t filter -I INPUT   -j DROP
iptables -t filter -I FORWARD -j DROP
iptables -t filter -I OUTPUT  -j ACCEPT

# acceptations supplémentaires
iptables -t filter -I INPUT      -m state --state ESTABLISHED,RELATED                          -j ACCEPT
iptables -t filter -I FORWARD    -m state --state ESTABLISHED,RELATED                          -j ACCEPT
iptables -t filter -I FORWARD               -p icmp                                            -j ACCEPT
# SSH DSI
iptables -t filter -I INPUT      -i eth0.10  -p tcp --dport 22 -s 10.20.0.0/16 -d 10.20.0.250   -j ACCEPT
# web
iptables -t filter -I FORWARD               -p tcp --dport 80                 -d 10.30.0.2     -j ACCEPT
# DNS
iptables -t filter -I FORWARD               -p udp --dport 53                 -d 10.30.0.1     -j ACCEPT

# DNAT
iptables -t nat    -I PREROUTING -i eth0.100 -p tcp --dport 80                 -d 194.167.110.1 -j DNAT --to-
destination 10.30.0.2

iptables -P <chaine> <action>  action par défaut
iptables –F      réinitialiser

# filtrage
iptables
  -t         filter                           table
  -I/A       <INPUT / OUTPUT / FORWARD>       chaine
 [-m state –state ESTABLISHED,RELATED]        connexion déjà établies
  -i         <interface source>               interface source
  -s         <IP source>                      source
  -d         <IP destination>                 destination
  -p         <udp / tcp / icmp>               protocole
  --dport    <ssh / www / ... / num>          port
  -j         <ACCPEPT / DROP / REJECT>        action

# NAT
iptables
  -t         nas                              table
  -I/A       <PREROUTING / POSTROUTING>       chaine
   [...]                                      règles identiques
  -j         DNAT --to-destination <IP:port>  entrant dans le réseau privé
  -j         SNAT --to-source <IP>            sortant du réseau privé
  -j         MASQUERADE                       sortant du réseau privé avec une IP source auto
```

`/etc/init.d/iptables save` Sauvegarde des règles

NAT retour fait implicitement, pas besoin de règle.