\# cat /etc/resolv.conf :search;
nameserver 193.47.194.7
nameserver 193.47.194.9
cat /etc/hosts
adresse IP Nom d'hôte
127.0.0.1 localhost
208.77.188.166 example.com
Pour initialiser le réseau après configuration, il faut faire :
/etc/init.d/networking start
Le fichier`/etc/networks`
Ilpermetd'affecter un nom logique à un **réseau**

```
localnet    127.0.0.0
foo-net 192.168.1.0
```

Le fichier`/etc/host.conf`
Ildonnel'ordredanslequel le processus de résolution de nomsesteffectué. Voici un exemple de cequel'onpeuttrouverdanscefichier :

```
orderhosts,bind
/etc/network/interfaces
auto lo eth0 eth1
iface lo inet loopback
iface eth0 inet static
address 192.168.90.1
netmask 255.255.255.0
network 192.168.90.0
broadcast 192.168.90.255
```

gateway 192.168.90.1
Pour ethernet DHCP : iface eth0 inetdhcp
Confdhcp server : /etc/dhcp/dhcpd.conf
optiondomain-name"monserveur.com" : le ou les noms nom de domaine correspondant
au réseau local
subnetDonne une idée au serveur *DHCP* de la topologie du réseau. Cette option ne
change pas les accès ou les attributions d'adresses.

subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.2 192.168.0.20;
optionrouters 192.168.0.1;
default-lease-time 600;
max-lease-time 7200;
}
Réserve une adresse *IP* fixe particulière un un certain client identifié par son adresse
MAC.
hostguest {
hardware ethernet 67:42:AB:E3:74:00;
fixed-address 192.168.0.3;
}
Installation de *NFS :*
Le fichier /etc/exports permet de déclarer les répertoires à partager.
"répertoire local" "liste des machines autorisées à se connecter avec les options collées entre
parenthèses"
exemple : /homeollinux(rw) station1(ro)\\
exportfs –a : après chaque modification
showmmount –e : pour afficher les répertoire partager par la machine local
Coté client :
Mount –t nfsadr_server_nfs :chemin_rep
mount -t nfs 192.168.105.2:/armor/plages /mnt/cotes -o ro
pour afficher les repertoire partager sur un serveur nfs : showmount –e adr_server_nfs
Connection aux repertoires partages au démarrage
/etc/fstabSyntaxe :ordinateur-distant:répertoire-distant répertoire-local nfs options 0 0
monhost:/armor/plages /mnt/cotes nfsauto,rw,user,soft 0 0
Routage :
Route add –net network gwaddr_getway
Exemple : route add -net 192.168.0.0/24 gw 112.65.123.3
La passerelle doit être configurée pour transmettre (ou *forwarder*) les paquets *IP* d'un
réseau à l'autre, ce qui se fait par la commande
echo 1 >/proc/sys/net/ipv4/ip_forward ou bien d'une manière permanente : /etc/sysctl.conf
net.ipv4.ip_forward=1 .on peut voir l'etat des route par la commande route –n
routeadd default gw 194.56.87.1 route par defaut utilisé pour accéder a internet
NAT : si la passerellese connecte à internet via son interface eth0, il suffit d'exécuter la commande suivante sur lapasserelle :
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
Toute machine du réseau localqui se connecte à internetvia cette passerelle aura alors l'adresse *IP* de la passerelle sur internetOn peut aussi donneraux machines du réseau local une autre adresse *IP* que l'on spécifie avec –to :
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 193.56.17.9.
FIREWALL :
INPUT : paquets entrants à destination de la machine et venant d'une autre machine ;
OUTPUT : paquets sortants venant de la machine et à destination d'un autre machine ;
FORWARD paquets venant d'un autre machine et à destination d'une troisième machine
lors de l'utilisation de la machine comme passerelle pour le routage.
Pour afficher une chaine : iptables –t table –L chaine (table par defaut : filter)

Ajouter une régle : iptables –t table [-A|-D|-I n|-R n] chaine régle
A : Add, D : delete , I :inserer la regle a la position n , R : remplacer la n par cette regle
Dans une régle : -s : addr source, -d : adr_dest, -p : protocole, --sport : port source , --dport : port dest, -j action (ACCEPT | DROP | REJET)
Iptables –F chaine : supprimer tous les régle de la chaine.
Ex : iptables –A INPUT –p tcp –s 192.168.1.0/24 --dport 22 –j ACCEPT
-Pour changer la politique : iptables –P chaine politique (ACCEPT | DROP)
Dans une règle : -i interface d'entrée (INPUT|FORWARD) , -o interface sortie (OUTPUT | FORWARD)
Si on veut utiliser une liste de port ,il faut charger un module (multiport)
Ex : iptables –A INPUT –p tcp –m multiport –dport 80,443 –j ACCEPT
Type de packet ping : echo-request , echo-reply
EX : inderdit B de ping sur A ; sur la machine A iptables –A INPUT –p Icmp-type echo-reply –j DROP ; dans ce cas la politique par défaut sur A : pour INPUT est DROP , pour OUTPUT est ACCEPT
Si la politique par défaut sur A de INPUT est ACCEPT alors : iptables –A INPUT –p icmp --icmp-type echo-request –j DROP
Le package iprange permet de définir un intervalle des adr_ip
Ex : iptables –A INPUT –m iprange --src-range 192.168.1.10 --dst-range 192.168.1.20 –j ACCEPT .
Pour ajouter une chaine : iptables –N nom , iptabes –N log_http
Pour envoyer le trafic vers log_http : iptables –A INPUT –p tcp –dport 80 –j log_http . avant d'accepter on ajout une ligne dans le fichier log «iptables accept ». iptables –A log_http –j ACCEPT.
Etat de packet : NEW | ESTABLISHED | RELATED (on relation avec une connexion déjà établit) (ftp) INVALID : déférent des 3 précédents .
Politique par défaut INPUT=DROP ; on veut se connecter a ftp
Iptables –A INPUT –p ftp –m state -–state ESTABLISHED,RELATED –j ACCEPT.
Politique par défaut OUTPUT= DROP ; Autoriser ssh sur la machine
Iptables –A INPUT –p tcp --dport 22 -m state !- -state INVALID -j ACCEPT. ### iptables –A OUTPUT -p tcp –m state !- -state INVALID -j ACCEPT
Autorise RL → internet (FORWARD=DROP) iptables –A FORWARD –i eth1 -o eth2 -p tcp -m state !- - state INVALID –j ACCEPT .
Snat permet de changer l'adresse source de paquet ## table NAT
PREROUTING : avant l'opération de routage ## POSTROTING : après l'opération de routage ## OUTPUT ## INPUT
ROUTAGE : iptables –t nat –A POSTROUTING –j snat - - to-source @ip
Ou bien –j MASQUERAD (utiliser l'adresse de l'interface de sortie)
la passerelle peut rediriger les accès WEB (port 80) via son interface eth1 sur un serveur web situé sur une autre machine d'IP 192.168.0.5 sur le réseau local : iptables -A FORWARD -p tcp --sport www -j ACCEPT
iptables -A FORWARD -p tcp --dport www -j ACCEPT
iptables -t nat -A PREROUTING -p tcp -i eth1 --dport www -j DNAT –to 192.168.0.5.
Ssh : secure shell # # configuration du serveur ssh : /etc/ssh/sshd_config
Port 22 (on peut ecouter sur plusiers port a la fois)
Fichier log : /var/log/secure
Client ssh : les informations spécifiques a l'utilisateur se trouve dans ~/.ssh
Le fichier de configuration du client : /etc/ssh/ssh_config
Option de ssh : -l : login , -1 ou -2 : version, - p : port distant
Ex : ssh –vv -l user –p port –(1|2) hote ## ssh user@hote
3 methodes d'autentification ### 1/ par mot de passe : c'est la plus simple
Pour une premiere connexion il vous demande si le fingerprint de la clé public présentée par le serveur est bien le bon , si c'est le cas répondez par yes , alors la clé public du serveur est alors rajoutée au fichier ~/.ssh/known_hosts.
2/Authentification par clé : a/ générer un couple de clé
Ssh-keygen -t dsa (algo dsa) ssh-keygen –t rsa (algo rsa)
Par la suite une passephrase nous est demandée (sert a crypter la clé privée) par défaut la clé privé est stocké dans ~/.ssh/id_dsa et la clé public dans ~/.ssh/id_dsa.pub #### b/ autoriser votre clé pulic : pour cela il suffit de copier votre clé public dans le fichier ~/.ssh/authorized_keys sur la machine sur laquelle vous voulez vous connecter a distance

La commande suivante permet de réaliser cette opération :
scp .ssh/id_dsa .pub root@192.168.135.1: /root/.ssh/dsa_root.pub
le fichier est maintenant copier sur la machine , il reste a inclure la clé dans le fichier /$HOME/.ssh/authorized_keys ## sur la machine distant cd .ssh cat dsa_root.pub >> authorized_keys ## on peut maintenant se connecter sans mot de passe il faut juste fournir le passephrase.
3/ Authentification sans mot de passe
Ssh-agent : ce programme tourne en tache de fond et garde la clé en mémoire. La commande ssh-add permet donné sa clé a ssh-agent
Pour tuer l'agent ssh : ssh-agent –k .
Scp hote_d_ou_je_veux_copier : source_copie hote_destination : cible
L'option –r de copier un répertoire
Sftp user@host : secure ftp
----------FTP----------
Sereur ftp : vsftpd ## le fichier de configuration /etc/vsftpd.conf
Anonymous-enable = yes : permet les utilisateur anonyme a se connecter sur le serveur (login : anonymous , password : chaine vide)
Local_enable = yes : authorisé les utilisateurs local a se connecter
Write-enable = yes : autorise l'upload
Par défauts les utilisateurs /etc/ftp/users ne sont pas autorisé a se connecter
Coté client : le fichier .netr c automatise la connexion a un serveur ftp
Machine @ip login nom_user passwd password
Mkdir ~ftp/pub ## chown ftp :ftp ~ftp/pub
~ftp uniquement root est autorisé pour faire de modif , si on donne des autorisation W pour anonymous alors le système bloque anonymous de se connécter avec ftp sur ~ftp (anonymous fait des upload sur ~ftp/pub)
Chroot-list-enable= yes
Chroot-list-file=/etc/vsftp.chroot_list
Les utilisateur qui figure dans la liste sont chrooté
Chroot-local-user=yes les utilisateur de la liste ne sont pas chrooté .
---------- user-list-enable=yes ## user-list-file=/etc/vsftp.user_list
Les users qui figure dans la liste ne sont pas autorisé a se connecter
----------chroot-list-enable =yes ## chroot-list-file = fichier (contient la liste des utilisateurs chrooté) si chroot-local-user=yes le fichier contient les users non chrooté.
----------user_list_deny=yes (liste des users non autorisé)
Local_enable=yes ## userlist=yes ## userlist-file=fichier (contient la liste des users ont le droit de se connecter) ## Rq : le mot de passe n'est pas demandé pour les users non autorisé au contraire de /etc/ftpusers liste des users non autorisé mais le mot de passe est demandé.
Au point de vue sécurité il faut interdire les users locaux a se connecté a ftp
Userlist-deny=no ## local-enable=yes ## fichier userlist vide ;; on crée des users virtuels ;; création d'un fichier qui contient la liste des utilisateurs virtuels /etc/login.txt Nom_user01 \n password \n Nom_user02\n password
A la fin du fichier on ajoute une ligne vide ## dans le fichier /etc/vsftpf on ajoute les lignes suivant : local-enable=yes ## anonymous-enable=no## guest-enable=yes ## pam-service-name=/etc/pam.d/vsftpd2.
Vi /etc/pam.d/vsftpd2 (pour chaque application on peut faire des authentifications différents)
Auth required /lib/security/pam_userdb.so db=/etc/login
Account required /lib/security/pam_userdb.so db=/etc/login
#db-load -T –t hash -f /etc/login.txt /etc/login . db
Chmod 600 /etc/login . db