Module de Réseau

TD n°14 : Interfaces, routage et filtrage

Christian Toinard, Armen Petrossian, Hugo Lemarchant, Baptista Katapi, Benjamin Turquet

Le but de ce TD est d'étudier les interfaces réseaux sur une machine UNIX. Afin de procéder aux manipulations, le privilège root sera nécessaire sur la machine. Vous n'avez bien sûr pas accès aux droits administrateurs sur les machines de l'INSA. Nous utiliserons alors deux machines virtuelles sous Debian 6.0.7 grâce au logiciel VMware que nous appellerons respectivement VM1 et VM2, pour plus de compréhension.

1. Lancez VMware puis démarrez les deux machines virtuelles et connectez vous avec l'identifiant root muni du mot de passe azerty.
Depuis la fenêtre : dans l'onglet « Home » : « Open a Virtual Machine ».
Depuis la barre d'outils : « File » puis « Open… »

Rendez-vous ensuite dans le dossier contenant l'image de la machine virtuelle et sélectionnez le fichier .vmx correspondant.
2. Lister les interfaces réseaux disposant d'une adresse IP.
Donner les adresses IP de VM1 et VM2.

/sbin/ifconfig -a

3. Vérifier que leurs adresses IP sont bien sur le sous-réseau d'une interface virtuelle définie par VMware.
Donner l'adresse de ces sous-réseaux.

4. Supprimer la route par défaut de VM1 et ajouter une nouvelle route par défaut vers VM2.

route del default

route add default gw X.X.X.X

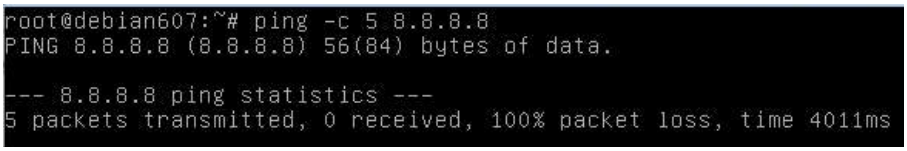
Vérifier que la nouvelle passerelle répond correctement.

ping X.X.X.X

```
root@debian607:~# ping -c 3 192.168.199.128
PING 192.168.199.128 (192.168.199.128) 56(84) bytes of data.
64 bytes from 192.168.199.128: icmp_req=1 ttl=64 time=6.22 ms
64 bytes from 192.168.199.128: icmp_req=2 ttl=64 time=0.275 ms
64 bytes from 192.168.199.128: icmp_req=3 ttl=64 time=0.316 ms

--- 192.168.199.128 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.275/2.270/6.220/2.793 ms
```

5. Vérifier que VM1 ne peut pas accéder à 8.8.8.8 (machine DNS de google)



6. Configurer VM2 comme routeur (c'est-à-dire que VM2 doit faire du « store and forward »). Vérifier que VM1 peut désormais accéder à 8.8.8.8.

echo 1 > /proc/sys/net/ipv4/ip_forward

7. Lancer WireShark sur la machine physique.
Faire une capture avec WireShark sur l'interface réseau VMware de la machine hôte qui permet à la machine virtuelle d'accéder à Internet.
Donner l'adresse IP de l'interface de l'hôte avec laquelle se fait la capture.
Justifier votre choix parmi les deux interfaces VMware de l'hôte.

8. Faire une capture d'écran de la capture WireShark et commenter les adresses source et destination.

9. Vous allez maintenant demander à VM2 de faire de la translation d'adresse (c'est-à-dire que l'adresse source est remplacée par l'adresse de VM2 sortant sur Internet).

iptables -t nat -A POSTROUTING -o interfaceX -j MASQUERADE

10. Lancer un ping depuis VM1 vers 8.8.8.8 et faire une capture sur l'hôte de ce qui circule avec WireShark.
En donnant une capture d'écran de WireShark, justifier que la translation a bien eu lieu.

11. Désormais, nous allons interdire tout le trafic entrant sur VM1 grâce aux commandes iptables suivantes. Commenter le rôle de chacune de ces commandes

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT

12. Utiliser SSH entre les deux machines virtuelles pour montrer que SSH marche dans un sens mais pas dans l'autre. Faire une trace des commandes SSH.

ssh root@*ip-machine-distante*