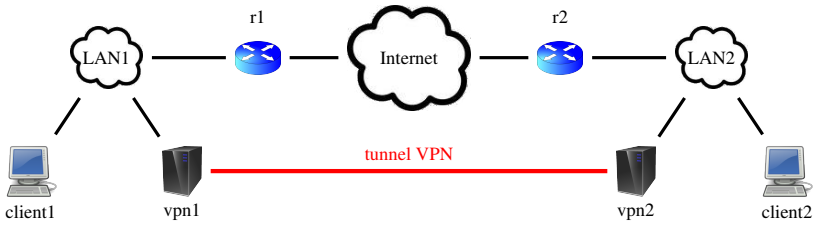
Travaux dirigés

Exercice 1 — Tunnels VPN

On considère dans cet exercice le réseau de la figure ci-dessous :



On considère dans un premier temps que le tunnel VPN est un tunnel de niveau 2 mis en œuvre avec openvpn. Les deux LAN forment le réseau 10.0.0.0/8.

Depuis le client1 (10.1.0.1), on exécute la commande ci-dessous (10.2.0.1 étant l'IP de client2) :

```
$ ping -c1 10.2.0.1
```

La table ARP du client1 est vide. Les tables ARP de tous les autres hôtes et routeurs sont remplies (i.e., elles contiennent toutes les adresses MAC de leur réseau).

Q 1.1 Donnez l'architecture logique du réseau.

Q 1.2 Quelles seront les messages échangés pour que la demande d'écho envoyée par la commande arrive à destination de client2 ?

Q 1.3 Donnez la structure du (des) message(s) transmis par le serveur VPN en faisant apparaître les différents en-têtes et en indiquant les adresses et numéros de port apparaissant dans ces en-têtes.

On considère maintenant que le tunnel est de niveau 3. Le LAN1 a pour adresse 10.1.0.0/16 et le LAN2 a pour adresse 10.2.0.0/16. On exécute la même commande depuis client1.

Q 1.4 Donnez l'architecture logique du réseau.

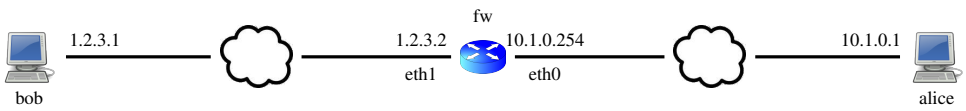
Q 1.5 Quelles seront les messages échangés pour que la demande d'écho envoyée par la commande arrive à destination de client2 ?

Q 1.6 Donnez la structure du (des) message(s) transmis par le serveur VPN en faisant apparaître les différents en-têtes et en indiquant les adresses et numéros de port apparaissant dans ces en-têtes.

Q 1.7 Donnez des avantages et inconvénients pour chaque niveau de tunnel (2 ou 3).

Exercice 2 — Règles de filtrage

On considère dans cet exercice le réseau de la figure ci-dessous :



On exécute sur le routeur fw le script ci-dessous.

```
1  #!/bin/bash
2  iptables -P FORWARD ACCEPT
3  iptables -P INPUT   DROP
4  iptables -P OUTPUT  DROP
5  iptables -A INPUT  -p icmp -i eth0 -j ACCEPT  # -i = input interface
6  iptables -A OUTPUT -p icmp -o eth0 -j ACCEPT  # -o = output interface
7  iptables -A INPUT -p tcp --dport 22 -i eth0 -j ACCEPT
8  iptables -A INPUT -j DROP
9  iptables -A FORWARD -i eth1 -p tcp --dport 80 -j ACCEPT
10 iptables -A FORWARD -i eth0 -p tcp --sport 80 -j ACCEPT
11 iptables -A FORWARD -p tcp -j DROP
```

Q 2.1 Une fois le script exécuté, indiquez, pour chaque commande ci-dessous si elle fonctionne ou pas. Justifiez en indiquant les lignes du script qui vous permettent de conclure. On supposera que toutes les conditions sont remplies par ailleurs pour que la commande fonctionne (p.ex., l'hôte cible répond bien aux messages ping, un serveur un écoute sur l'hôte cible sur le port indiqué dans la commande nc, ...).

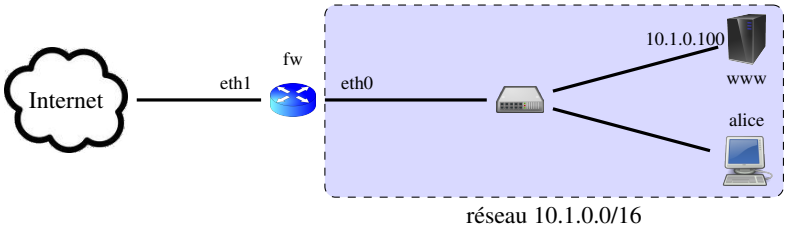
(a) Sur alice : `ping 10.1.0.254`
(b) Sur alice : `echo salut | nc 10.1.0.254 22`
(c) Sur alice : `echo salut | nc 1.2.3.1 80`
(d) Sur bob : `ping 1.2.3.2`
(e) Sur bob : `ping 10.1.0.1`
(f) Sur bob : `echo salut | nc -u 10.1.0.1 1234` (`-u` = mode UDP)
(g) Sur bob : `echo salut | nc 10.1.0.1 80`
(h) Sur bob : `echo salut | nc 10.1.0.1 22`
(i) Sur fw : `echo salut | nc 10.1.0.1 80`

Q 2.2 Quelle ligne est inutile dans ce fichier (i.e., elle peut être supprimée sans conséquence) ? Justifiez.

Q 2.3 Le script est-il équivalent si l'on intervertit les lignes 10 et 11 ? Justifiez.

Exercice 3 — Filtrages stateful/stateless

On considère dans cet exercice le réseau de la figure ci-dessous :



On souhaite configurer le pare-feu fw pour protéger le réseau 10.1.0.0/16. Voici les contraintes du cahier des charges :
(C1) Le pare-feu peut être contacté par les hôtes du réseau 10.1.0.0/16 mais pas par l'extérieur (Internet).
(C2) Tous les hôtes du réseau 10.1.0.0/16 peuvent communiquer sans contrainte avec l'extérieur.
(C3) L'hôte www est un serveur web accessible depuis l'extérieur sur le port TCP/80.
(C4) Tout autre échange initié depuis l'extérieur est bloqué.

Q 3.1 Proposez une suite d'instructions iptables permettant de remplir ces contraintes. On pourra utiliser le module state d'iptables pour mettre en place un filtrage stateful.

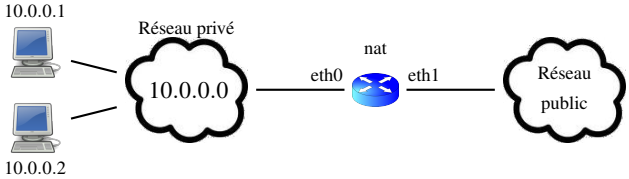
Q 3.2 Comment pourrait-on mettre en œuvre les contraintes *C3* et *C4* avec un filtrage stateless ? (On considèrera le cas de TCP uniquement.)

Le balayage ACK de nmap consiste à envoyer un paquet TCP avec le bit ACK activé (et celui-ci seulement) vers le port d'un hôte cible. Puis :
— si nmap reçoit en réponse un paquet RST, il conclut que le port est *non filtré* ;
— et s'il ne reçoit aucune réponse, il conclut que le port est *filtré*.

Q 3.3 Le balayage ACK est surtout intéressant car il permet de déterminer le type de pare-feu (staful ou stateless) entre nmap et la cible. Pourquoi ?

Exercice 4 — Translation d'adresse

On considère dans cet exercice le réseau de la figure ci-dessous :



L'hôte nat joue le rôle de passerelle NAT entre le réseau privé et le réseau public. L'IP publique de son interface eth1 qui la relie au réseau public est 1.2.3.4. On suppose que la passerelle est en mode NAPT (i.e., elle fait de la translation d'adresse et de port). Elle attribue les numéros de port séquentiellement dans l'intervalle [10000,20000].

La passerelle NAT reçoit successivement les paquets suivants :

1. eth0 : 10.0.0.1:5471 → 2.3.4.5:22

2. eth0 : 10.0.0.2:10887 → 3.4.5.6:80
3. eth1 : 2.3.4.5:22 → 1.2.3.4:10000
4. eth1 : 5.6.7.8:443 → 1.2.3.4:10002
5. eth1 : 2.3.4.5:22 → 1.2.3.4:10001

Q 4.1 Pour chaque paquet, indiquez :
   (a) si le paquet sera routé ou détruit
   (b) et s'il est routé, la translation faite par la passerelle et
   (c) le contenu de la table NAT après routage.

Q 4.2 Dans le cas d'une connexion TCP, à quel moment la passerelle peut-elle retirer une ligne de sa table ?

Q 4.3 Supposons que l'hôte 10.0.0.2 soit un serveur web écoutant sur les 80 (http) et 443 (https). Que faut-il faire pour rendre ce serveur web accessible depuis le réseau public ?

Q 4.4 Donnez les règles iptables permettant de mettre en place la translation d'adresse (y compris celle énoncée dans la question précédente). On ne mettra pas en place de translation de port mais uniquement d'adresse.