Filtrage

Prénom : **Nom :** **Groupe :**

1 Gestion du réseau virtuel

Le réseau virtuel utilisé lors de ce TP a été réalisé avec **NEmu** (Network Emulator), un émulateur de réseaux virtuels distribués et dynamiques utilisant QEMU (logiciel d'émulation de machines). NEmu a été conçu par des membres du LaBRI, pour plus d'informations consulter le site `http://nemu.valab.net`.

1.1 Création des machines, lancement de NEmu

Après vous êtes positionné dans votre sous-répertoire ~`/VMs/VNET`, exécutez le script `./restore filtrage.tgz`, puis le script `./vnet filtrage`.

Deux commandes de NEmu vous seront utiles : `quit()` et `save()`.

A chaque machine virtuelle est associée une fenêtre graphique où tourne un terminal. Pour travailler dans une machine virtuelle, il faut positionner la souris dans la fenêtre associée à la machine, puis clic sur le bouton droit de la souris.

La souris peut être" bloquée" (par un clic sur le bouton gauche de la souris) dans une fenêtre associée à une machine virtuelle, pour "débloquer" la souris, il faut taper "Ctrl Alt" dans la fenêtre incriminée.

1.2 Information sur les machines

Les commandes Linux nécessaire au TP sont réalisables uniquement par l'administrateur des machines (login : root, passwd : plop).

Les machines virtuelles utilisent la distribution Linux, `Debian`.

L'éditeur de texte `jed` est préinstallé.

La commande `startx` permet de remplacer le terminal textuel par un environnement graphique basique). Un clic sur un bouton de la souris permet d'accéder au menu :

1. `exit` pour quitter l'envionement graphique,
2. `application -> reseau -> surveillance -> Wireshark` pour lancer Wireshark, renifleur et analyseur de trames réseau.

1.3 Arrêt d'une machine virtuelle

Dans le terminal associé à la machine, exécutez la commande `halt` avant de détruire la fenêtre associée à la machine (c'est à dire de cliquer sur l'icône de destruction de la fenêtre).

1.4 Sauvegarde des machines et restauration des machines

Sauvegarde des machines : après l'arrêt propre de toutes les machines, exécutez la commande `save()` de NEmu. Le fichier `~/filtrage.tgz` contiendra la sauvegarde des machines virtuelles. Pour finir, quittez NEmu en tapant la commande `quit()` de NEmu.

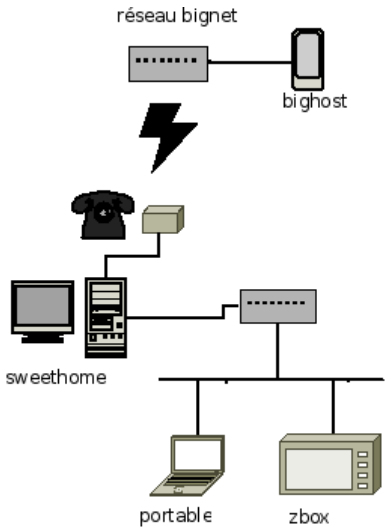
Restauration des machines : après s'être positionné dans votre sous-répertoire `~/VMs/VNET`, exécutez le script `./restore ~/filtrage.tgz`, avant de créer le réseau via le script `./vnet filtage`.

2 Objectif du TP : Filtrage

Vous avez pris un abonnement chez un FAI (Fournisseur d'Accès à Internet) ici `bignet.com`. Vous cherchez tout d'abord à protéger le poste de travail familial "`sweethome`" des attaques éventuelles.

La famille s'équipant, apparaît ensuite le besoin de *partager la connexion Internet* (le FAI ne fournit qu'une ligne avec une adresse IP) avec divers équipements (ici un portable et une console de jeu `zbox`), que l'on raccorde par un switch ; le poste de travail - muni d'une seconde carte réseau - agissant comme routeur.

Et ensuite, vous voulez que le jeu en réseau qui tourne sur le serveur web de la Zbox soit accessible de l'extérieur...



Mémento `iptables`

Tables et chaînes prédéfinies

– La table par défaut (`filter`) contient les chaînes prédéfinies `INPUT`, `OUTPUT` et `FORWARD`.
`INPUT` traite les paquets IP dont la *destination* est locale (une interface de la machine).
`OUTPUT` traite les paquets IP dont la *provenance* est locale.
`FORWARD` traite les paquets IP en transit.
– La table `nat` contient les chaînes `PREROUTING` et `POSTROUTING`.
`PREROUTING` traite les paquets IP dont la destination est locale ainsi que les paquets IP en transit. Les règles de type `PREROUTING` sont appliquées *avant les autres règles.*
`POSTROUTING` traite les paquets IP dont la source est locale ainsi que les paquets IP en transit. Les règles de type `POSTROUTING` sont appliquées *après les autres règles.*

Commandes IPTABLES

Voir les règles	`iptables -t filter -L ; iptables -t nat -L`
Définir une politique par défaut	`iptables -t filter -P INPUT DROP` `iptables -P INPUT DROP`

Conditions IPTABLES

Interface entrante	`--in-interface eth0`	`-i eth0`
Interface sortante	`--out-interface eth1`	`-o eth1`
Adresse IP destination	`--destination 10.1.1.45`	`-d 10.1.1.45`
Adresse IP source	`--source 10.1.1.0/24`	`-s 10.1.1.0/24`
Protocole	`--protocol tcp`	`-p tcp`
Port (avec TCP ou UDP)	`--source-port 53`	`--sport 53`
	`--destination-port http`	`--dport http`
Saut vers une cible (action)	`--jump ACCEPT` `-j ACCEPT`	
État de la connexion (avec TCP)	`--protocol tcp -m state --state NEW` `-p tcp -m state --state ESTABLISHED,RELATED`	

Traduction d'adresse (NAT)

Masquerading (SNAT)	`iptables -t nat -A POSTROUTING` `-s 10.1.1.0/24 -o eth1 -j MASQUERADE`
Redirection (DNAT)	`iptables -t nat -A PREROUTING -i eth0 -p tcp` `--dport 80 -j DNAT -d 10.1.1.2:80`

3 Découverte du réseau

Au départ, le script de simulation de réseau `filtrage` ne lance que les deux machines `sweethome` et `bighost`.

1. Connectez-vous sur les deux machines. Faites un plan du réseau avec les adresses IP utilisées, les adresses des réseaux (avec leur masque).

2. Donnez la table de routage de `bighost` :

Adresse du réseau	Nom de l'interface réseau	Passerelle

3. Précisez le(s) serveur(s) de noms utilisé(s) par `bighost` :

4. Donnez la table de routage de `sweethome` et précisez le(s) serveur(s) de noms utilisé(s).

Adresse du réseau	Nom de l'interface réseau	Passerelle

5. Précisez le(s) serveur(s) de noms utilisé(s) par `sweethome` :

6. Étudiez le domaine DNS géré par `bighost` (n'hésitez pas à regarder les fichiers de configuration !) :
   (a) quel est son nom ?

(b) quel est le nom des fichiers où sont stockés les données ?

(c) quelle est l'adresse IP de `www.bignet.com` ?

(d) quelle est l'adresse IP de `dns.bignet.com` ?

(e) quelle est l'adresse IP de `poste.bignet.com` ?

(f) quelle est l'adresse IP de `autre.bignet.com` ?

7. Étudiez le domaine DNS géré par `sweethome` (n'hésitez pas à regarder les fichiers de configuration !) :
   (a) quel est son nom ?
   (b) quel est le nom des fichiers où sont stockés les données ?

   (c) quelle est l'adresse IP de `sweethome.localdomain` ?
   (d) quelle est l'adresse IP de `dns.localdomain` ?
   (e) quelle est l'adresse IP de `zbox.localdomain` ?
   (f) quelle est l'adresse IP de `portable.localdomain` ?
8. Listez les noms associés à chaque adresse IP de `sweethome`.

Adresse IP	Nom

9. Sur le poste de travail, `sweethome`, le script `/etc/init.d/firewall` contient les règles de filtrage activées à chaque démarrage.
   – Quand est-il lancé exactement ?
   – Pour quelle raison choisit-on ce moment précis ? (les points d'entrée pour répondre à ces questions sont `/etc/inittab` et `/etc/rc*.d`)

Le script `/etc/init.d/firewall` sera développé durant la suite du TP. Après chaque modification du script ; il faut Relancer le service (commande : `/etc/init.d/firewall restart`)

4 Protection de `sweethome`

1. Quels services tournent sur `bighost` et `sweethome` (`ssh`, `dns`, `web`...) ? **Indices :** utilisez la commande `netstat -a`.

Hôtes	services

2. Depuis `sweethome`, tentez un ping vers `bighost.bignet.com`.
   Regardez le fichier `/etc/init.d/firewall` et expliquez votre échec. Depuis `bighost.bignet.com`, tentez un ping vers `poste.bignet.com`. Expliquez votre échec.

3. Ajoutez une ou plusieurs règle(s) `iptables` au fichier `/etc/init.d/firewall` pour autoriser **seulement** les pings sortants et entrants. Vérifiez.
   Donnez les règles ajoutées :

4. Sur `bighost`, créez un utilisateur `boss`, et donnez-lui un mot de passe :
   ```
   useradd -m boss
   passwd boss
   ```
5. Depuis `sweethome`, tentez une connexion ssh sur `bighost` sur le compte de boss : `ssh boss@bighost.bignet.com`. Expliquez votre échec.

6. Dans `/etc/init.d/firewall`, ajoutez une ou des règle(s) pour autoriser **seulement** les connexions `ssh` sortantes. Vérifiez.
   Indices : il faut déterminer le protocole transport utilisé par ssh (TCP ou UDP), le port destination des datagrammes sortants et le port source des datagrammes entrants.

Donnez les règles ajoutées :

7. Depuis `sweethome`, tentez des connexions web vers `bighost.bignet.com` (utilisez la commande `wget http://bighost.bignet.com`. Si vous êtes en mode graphique, vous pouvez utiliser la commande `firefox http://bighost.bignet.com`. Expliquez votre échec.

8. Dans `/etc/init.d/firewall`, ajoutez une ou des règle(s) pour autoriser **seulement** les connexions `http` sortantes. Vérifiez.
   Indices : il faut déterminer le protocole transport utilisé par HTTP. Il faut aussi déterminer le port destination des datagrammes sortants, et le port source des datagrammes entrants.
   Donnez les règles ajoutées :

9. Sur `sweethome`, créez un utilisateur à votre nom.
10. Essayez `ssh` vers `poste.bignet.com` depuis `bighost`. Expliquez votre échec.

11. Dans `/etc/init.d/firewall`, ajoutez une ou des règle(s) pour autoriser **seulement** les connexions `ssh` entrantes. Vérifiez.
    Indices : Il faut aussi déterminer le port source des datagrammes sortants, et le port destination des datagrammes entrants.
    Donnez les règles ajoutées :

12. Expliquez la commande `iptables` suivante : `iptables -A INPUT -i eth1 -j ACCEPT`

5 Extension du réseau, SNAT

1. Exécutez `startlocal()` dans la console NEmu.
2. Faites un plan du nouveau réseau avec les adresses IP utilisées, les adresses des réseaux (avec leur masque).

3. Donnez la table de routage de `portable` :

Adresse du réseau	Nom de l'interface réseau	Passerelle

4. Précisez le(s) serveur(s) de noms utilisé(s) par `portable` :

5. Dans quel fichier, cette information est-elle enregistrée ?

6. Donnez la table de routage de `zbox` :

Adresse du réseau	Nom de l'interface réseau	Passerelle

7. Précisez le(s) serveur(s) de noms utilisé(s) par `zbox` :

8. Depuis `portable`, tapez la commande `nslookup bighost.bignet.com`. Que fait cette commande ?
   À quel serveur sont transmises les requêtes DNS de `portable` ?

   `sweethome` connaît il l'adresse IP associée à `bighost.bignet.com` ?

9. Ecrivez les lignes du fichier de configuration `/etc/bind/named.conf.options` de `sweethome` expliquent le résultat de la commande `nslookup bighost.bignet.com` ? (pour en être

absolument sûr, commentez les 3 lignes, relancez le serveur DNS de `sweethome` et réessayez).

10. Depuis `portable`, effectuez des `ping`s vers `bighost`, avec son nom, son adresse, etc. Lors d'un `ping`, une requête ICMP est envoyée au poste distant, qui doit répondre en envoyant un paquet ICMP à l'émetteur du `ping`. Espionnez les trames recues sur les deux interfaces réseau du poste `sweethome`. La requête ICMP est-elle arrivée à `bighost` ? Expliquez.

11. Ajoutez dans le fichier `/etc/init.d/firewall` de `sweethome` une règle ou plusieurs règles assurant que les requêtes ICMP venant du réseau domestique soit transmises à `bighost`. Vérifiez.
Donnez les règles ajoutées :

12. Espionnez les trames recues sur les deux interfaces réseau du poste `sweethome`.`bighost` répond-il à la requête ICMP ? Expliquez.

13. Modifiez le fichier `/etc/init.d/firewall` de `sweethome` (et lancez le parefeu) pour mettre en route le *masquerading*. Faites des tests : depuis `portable`, effectuez plusieurs `ping`s vers `bighost`, avec son nom, son adresse, etc. Expliquez les résultats.
Donnez les règles ajoutées au fichier `/etc/init.d/firewall` de `sweethome` :

14. Listez les trames contenant un paquet ICMP reçues et émise par `sweethome`, lors de l'exécution de la commande `ping -c1 bighost.bignet.com` à partir de `portable`.

adresse physique de l'émetteur	adresse physique du destinataire	adresse logique de l'émetteur	adresse logique du destinataire

6 Redirection de services, DNAT

1. Vérifiez qu'un serveur web tourne bien sur `zbox`, et que vous pouvez le consulter depuis les machines du réseau familial (et bien sûr pas depuis l'extérieur)
2. Complétez et mettez en service les règles de filtrage concernant le DNAT : il s'agit de rediriger les paquets destinés au port 80 de `poste.bignet.com` vers `zbox` port 80 (modification d'adresse du destinataire), et qu'au retour, les réponses semblent provenir de `sweethome` (et non de `zbox`). Testez sur `bighost`.
   Donnez les règles ajoutées au fichier `/etc/init.d/firewall` de `sweethome` :