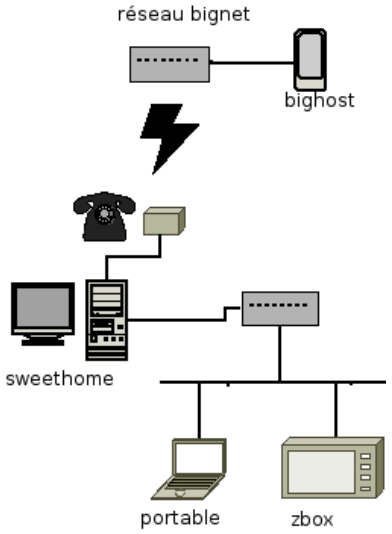
Filtrage

Vous avez pris un abonnement chez un FAI[1] (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...



1 Préparation

Dans `/net/exemples/ASR4-Reseaux/FILTRAGE` se trouvent
- un script `installer-machines`, lancez-le pour copier les 4 machines virtuelles de l'exercice dans votre répertoire `/.cows`,
- un script `reseau-maison` simulant le réseau, copiez-le dans votre espace de travail.

2 Exploration

Au départ, le script de simulation de réseau `reseau-maison` ne lance que les deux machines `poste` et `exterieur`.

1. Connectez-vous sur chaque machine et faites un plan du réseau avec les adresses IP utilisées et les routes de chaque machine, le(s) serveur(s) de noms utilisé(s).
2. Recensez également les domaines gérés par les DNS (n'hésitez pas à regarder les fichiers de configuration !)
3. Sur le poste de travail `poste`, le script `/etc/init.d/firewall` contient les règles de filtrage activées à chaque démarrage. C'est ce script qu'il faudra développer pendant cet exercice. 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`)

[1] Fournisseur d'accès à Internet

3 Protection du poste

1. Déterminez quels services tournent sur `bighost` et `sweethome` (`ssh`, `dns`, `web`...) : utilisez `ps -aux`, `netstat -a`, regardez `/var/log/messages`,...
2. Depuis le poste de travail, tentez des connexions vers `bighost.bignet.com` : `ping`, `ssh` et web. (pour le web : `lynx http ://bighost.bignet.com`). Conclusions ?
3. Dans `/etc/init.d/firewall`, ajoutez une règle pour autoriser les connexions `ssh` entrantes. Relancez le script (`/etc/init.d/firewall restart`). Vérifiez, etc.

4 Extension du réseau, SNAT

1. Arrêtez proprement les machines virtuelles. Dé-commentez les deux lignes du script `reseau-maison` qui concernent le `portable` et la `zbox`. Relancez. Déterminez les adresses, routes et DNS utilisés.
2. Depuis le portable, tapez la commande `host bighost.bignet.com`. Conclusions ?
3. Et maintenant un ping vers `bighost`, avec son nom, son adresse.... Conclusions ?
4. Quelle ligne des fichiers de configuration (`/etc/bind/named.conf.*`) du DNS explique le comportement du DNS constaté ci-dessus ? (pour en être absolument sûr, mettez-la en commentaire, relancez le serveur, et réessayez).
5. Faites tourner un `tcpdump` sur chaque interface réseau du poste (option`-i ethN`) : pour cela, modifiez le fichier `/etc/inittab` et redémarrez le poste pour obtenir deux consoles.

   Étudiez attentivement les paquets IP qui passent lorsque le portable veut accéder par `ping` (ou `ssh`) à `bighost`. Quels paquets manquent et pourquoi ? (comparez avec un ping depuis le poste vers `bighost`)
6. Modifiez `/etc/init.d/firewall` (et lancez-le) pour mettre en route le *masquerading*. Quelle différence dans les paquets IP émis/reçus ?

5 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 `poste` (et non de `zbox`).

Memento `iptables`

Tables et chaînes prédéfinies

- La table par défaut (`filter`) contient les chaines prédéfinies `INPUT` (resp. `OUTPUT`) qui traîte les paquets IP dont la destination (resp. provenance) est celle d'une des interfaces de la machines, et `FORWARD` pour les autres.
- La table `nat` contient les chaînes `PREROUTING` et `POSTROUTING` des règles à appliquer avant/après celles d'`INPUT`, `OUTPUT`, `FORWARD`

Commandes IPTABLES

Voir les règles	`iptables -L ; iptables -t nat -L`
Définir politique par défaut	`iptables -P INPUT DROP`
Effacer règles d'une chaine	`iptables -F INPUT`
Créer une chaine	`iptables -N chaine`
Supprimer une chaine	`iptables -X chaine`
Ajouter une règle	`iptables -A matable -s 10.1.1.0/24 -j ACCEPT`

Conditions IPTABLES

Adresse IP (source, destination)	`-s 10.1.1.0/24` `-d 10.1.1.45`
Protocole	`-p tcp`
Port (avec TCP ou UDP)	`--source-port 53` `--destination-port 1024:10000`
Interface	`-i eth0 -o ppp0`
État de la connexion	`-m state --state ESTABLISHED,RELATED` `-m state --state NEW`
Saut vers une cible ou une table de règles	`-j ACCEPT` `-j matable`

Cibles prédéfinies : `ACCEPT`, `DROP`, `REJECT`

Traduction d'adresse (NAT)

Masquerading (SNAT)	`iptables -t nat -A POSTROUTING`
	`--source 10.1.1.0/24 -o eth1`
	`-j MASQUERADE`

Redirection (DNAT)	`iptables -t nat -A PREROUTING -i eth0`
	`-p tcp --dport 80 -j DNAT`
	`--to-destination 10.1.1.2:80`