U.F.R. Sciences de l'Homme et de la Société
Master MIASHS

TP réseau firewall

L'objectif de ce TP est de comprendre comment mettre en place un routeur pare-feu (*firewall*) entre un réseau privé et un réseau public (Internet) à l'aide des règles de filtrage disponibles dans le noyau Linux avec `iptables`.

Les éléments nécessaires : *masquerading* + iptables

Le camouflage IP (*IP Masquerading*)

Les paquets venant de réseaux privés (tels que définis dans la RFC 1918, c'est-à-dire 10.*.*.*, 172.16.*.* ou 192.168.*.*) ne sont **pas routables** (les routeurs sont configurés dans ce sens).

Mais grâce à une spécificité du noyau Linux, le *camouflage* ou *masquage* (*masquerading*), activé sur un hôte qui va jouer le rôle de passerelle, toutes les machines du réseau privé pourront accéder de manière invisible à Internet. La passerelle apparaîtra comme étant le seul système utilisant la connexion Internet.

Le **camouflage réécrit les paquets** lorsqu'ils passent par la passerelle, ce qui fait qu'ils semblent toujours provenir de la passerelle elle-même. Il réécrit ensuite les réponses afin qu'elles semblent venir du destinataire originel.

Pour plus d'informations sur l'IP masquerading, vous pouvez consulter :
http://www.e-infomax.com/ipmasq

Le filtrage des paquets

Le programme `iptables` sert à manipuler les règles de filtrage de paquets au niveau du noyau Linux. Il permet de configurer un pare-feu. `iptables` peut aussi être utilisé pour contrôler le camouflage (ce sera le cas pour ce TP).

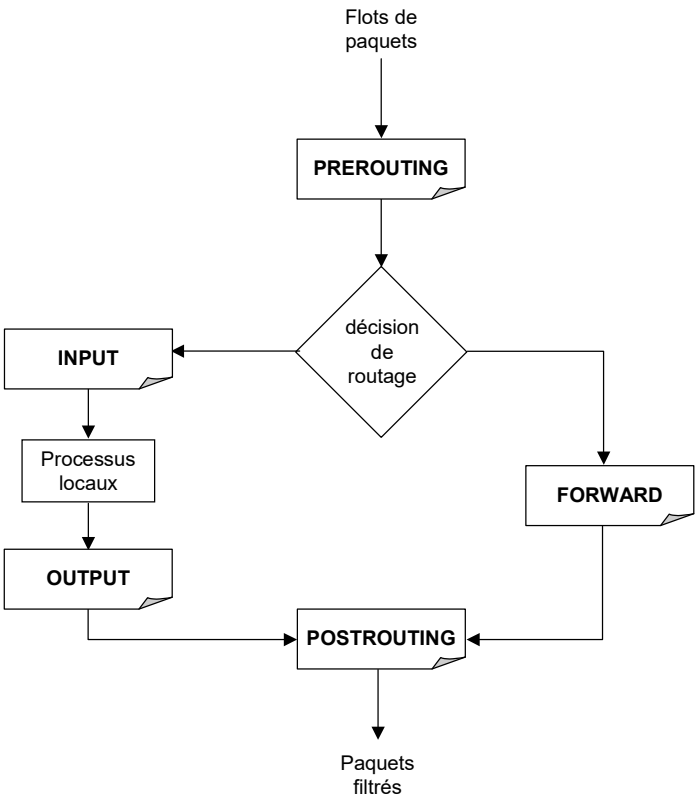
`iptables` ou `netfilter` (http://netfilter.samba.org) est apparu avec les noyaux 2.4.

Le noyau dispose de listes de règles appelées des **chaînes**. Les règles **sont analysées les unes à la suite des autres dans l'ordre de leur écriture**. Dès qu'une règle peut s'appliquer à un paquet, elle est déclenchée, et la suite de la chaîne est ignorée.

Les chaînes sont regroupées dans des **tables**. Il existe trois tables mais nous n'en utiliserons que deux :

- Table **NAT** (*Network Address Translation*) : elle est utilisée pour la translation d'adresse ou la translation de port. Deux types de chaînes s'appliquent à cette table :
  - PREROUTING
  - POSTROUTING

- Table **FILTER** : c'est la table par défaut. Elle contient toutes les règles de filtrage. Trois types de chaînes s'appliquent à cette table :
  - INPUT
  - OUTPUT
  - FORWARD



Un paquet rentrera toujours dans la machine via la chaîne PREROUTING et sortira toujours de la machine via la chaîne POSTROUTING.
Si le paquet doit être routé, il passera dans la chaîne FORWARD. Les chaînes INPUT et OUTPUT quant à elles serviront respectivement à placer des règles pour les paquets **destinés au** et **émis par** le firewall lui-même.

Chaque chaîne peut fonctionner selon trois politiques différentes :
- **ACCEPT** : tous les paquets sont acceptés.
- **DROP** : les paquets sont refusés sans notification à l'émetteur des paquets.
- **REJECT** : les paquets sont refusés mais avec notification à l'émetteur des paquets.

A l'aide des règles affectables à chaque chaîne, il est possible d'autoriser, de restreindre ou d'interdire l'accès à différents services réseaux, et ainsi modifier la politique de filtrage des paquets de chaque chaîne.
En fait, le filtrage de paquet fonctionne en analysant les **entêtes** des paquets. Si les données contenues dans l'entête d'un paquet correspondent à une règle alors la règle est appliquée (acceptation ou refus du paquet), sinon la règle suivante est examinée jusqu'à ce que toutes les règles de la liste affectée à chaque chaîne soient testées.

Pour plus de détails sur `iptables`, consultez la documentation disponible (`man iptables`, http://netfilter.samba.org).

Par défaut la table FILTER est vide et donc accepte tout. Aucune règle de translation d'adresse n'est présente par défaut.

Commandes utiles

- Configuration des adresses IP des interfaces par la commande `ifconfig`
  Chaque interface est identifiée par un nom :
  - *eth0* : première carte Ethernet
  - *lo* : *loopback* ou interface de bouclage

  Liste des interfaces réseau configurées :
  `ifconfig`

  Pour configurer une interface réseau :
  `ifconfig` `interface adresse_IP` **`netmask`** `masque_de_réseau` **`up`**

  Exemple :
  ```
  ifconfig eth0 192.168.10.1 netmask 255.255.255.0 up
  ```

- Configuration de la table de routage : la commande `route`

  Affichage de la table de routage :
  `route -n`

  Pour ajouter une entrée de réseau à la table de routage :
  `route add -net` `adresse_réseau_IP` **`netmask`** `masque_de_réseau` **`gw`** `adresse_routeur`

  Exemples :
  ```
  route add -net 127.0.0.0 netmask 255.0.0.0 dev lo
  route add -net 192.168.10.0 netmask 255.255.255.0 gw 192.168.1.1
  ```

  Pour ajouter un routeur par défaut à la table de routage :
  `route add default gw` `adresse_IP_routeur`

  Exemple :
  ```
  route add default gw 192.168.10.1
  ```

- Configuration des règles de filtrage : `iptables`

  Les opérations servant à gérer les chaînes entières (les 3 chaînes intégrées INPUT, OUTPUT et FORWARD ne peuvent pas être effacées) :

Créer une nouvelle chaîne utilisateur	`-N`	`iptables -N test`
Supprimer une chaîne utilisateur vide	`-X`	`iptables -X test`
Changer la police d'une chaîne intégrée	`-P`	`iptables -P FORWARD DROP`

Afficher les règles d'une chaîne ou de toutes les chaînes
sous forme numérique
	`-L`
	
	`-n`
		`iptables -L INPUT`
		`iptables -L`
		`iptables -nL`

Supprimer les règles d'une chaîne ou de toutes les chaînes	`-F`	`iptables -F INPUT` `iptables -F`

Les moyens pour manipuler les règles à l'intérieur d'une chaîne :

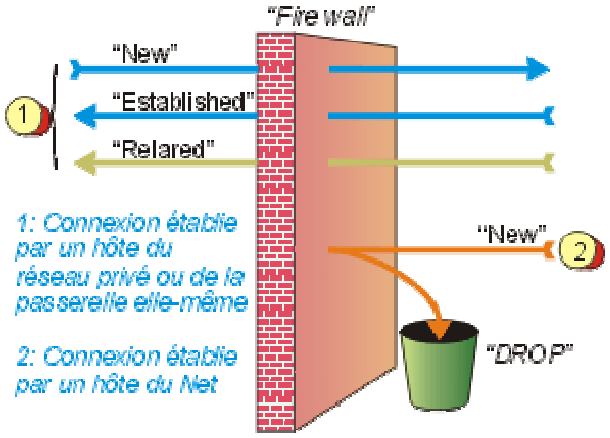
Ajouter une nouvelle règle à une chaîne	`-A`	`iptables -A INPUT -s 0/0 -j DENY`
Insérer une nouvelle règle à une position quelconque de la chaîne	`-I`	
Remplacer une règle à une position quelconque de la chaîne	`-R`	
Supprimer une règle à une position quelconque de la chaîne	`-D`	`iptables -D INPUT 1`
Supprimer la première règle vérifiée dans la chaîne	`-D`	`iptables -D INPUT -s 127.0.0.1 -p icmp -j DENY`

Les **adresses IP** source (option `-s` ou `--source`) et destination (option `-d` ou `--destination`) peuvent être spécifiées :

- en utilisant le nom complet, comme `prevert.upmf-grenoble.fr`
- en utilisant l'adresse IP, comme `195.221.42.159`
- en indiquant un groupe d'adresse IP, comme `195.221.42.0/255.255.255.0` (c'est-à-dire toutes les adresses du réseau 195.221.42.0) ou en notation condensé `195.221.42.0/24`
- en désignant n'importe quelle machine : `0.0.0.0/0` ou `0/0`

L'**état du paquet** peut être spécifié en utilisant l'option `--state` :

- `ESTABLISHED` : paquet associé à une connexion déjà établie
- `NEW` : paquet demandant une nouvelle connexion
- `INVALID`: paquet associé à une connexion inconnue
- `RELATED` : nouvelle connexion mais liée, idéal pour les connexions FTP



Le **protocole** peut être spécifié en utilisant l'option `-p` ou `--protocol`. Ce peut être un nom parmi ICMP, TCP ou UDP (en majuscule ou minuscule) ou le nombre correspondant au protocole (respectivement 1, 6 et 17 cf `/etc/protocols`).

Lorsque TCP ou UDP est spécifié, un argument supplémentaire indique le **port source** (option `--sport` ou `--source-port`) ou le **port destination** (option `--dport` ou `--destination-port`). Il peut s'agir

- d'un seul port, comme `--sport 80`
- de plusieurs ports (nécessite l'option `-m multiport`), comme `-m multiport --dport 137,139`
- d'un intervalle (inclusif) de ports indiqué par le caractère ":", comme `--sport 1024:65535`

La négation est spécifiée en utilisant le caractère "!".

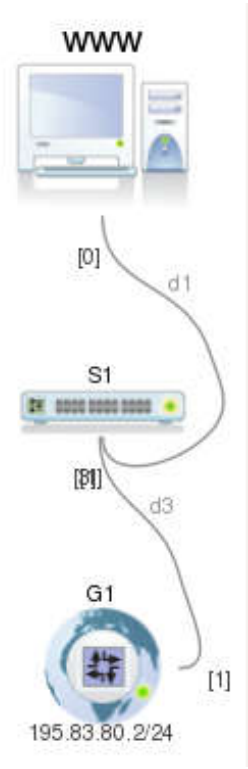
Pour les options restantes, consultez le manuel (`man iptables`).

Préambule

1) Copiez sur votre bureau le projet nommé `firewall.mar` qui se trouve sur la page Web

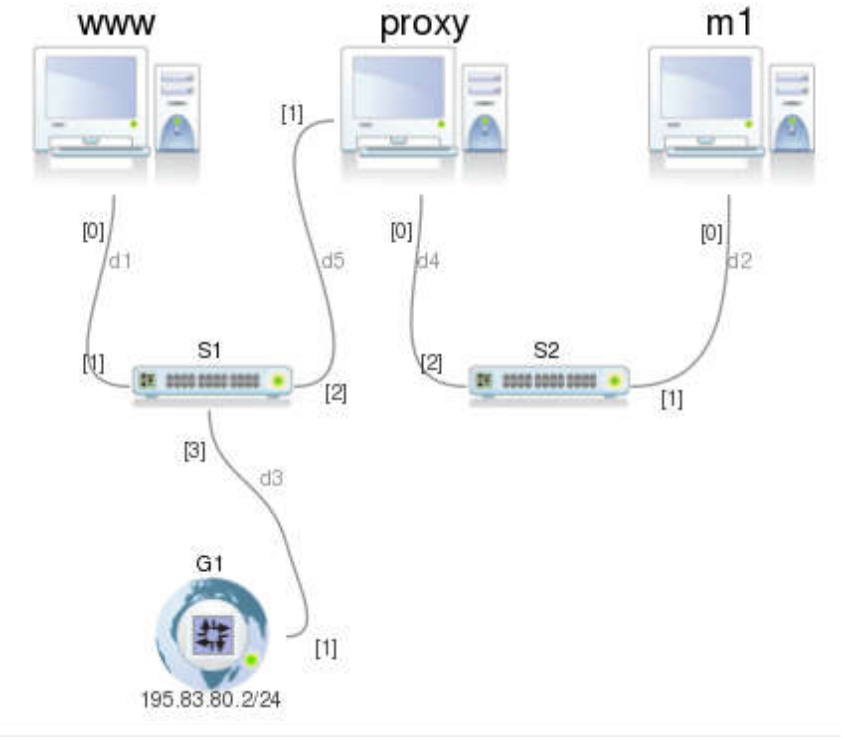
2) Lancez Marionnet

   Allez dans le menu Projet > Ouvrir, puis sélectionnez le projet que vous venez de copier. Vous devriez vous retrouver avec une machine nommée *www* et un équipement nommé *G1* dans le réseau public 195.83.80.0/24.



Configuration du réseau : étape ❶

1) Créez un réseau privé d'adresse 192.168.40.0/24, contenant 2 machines : un simple **client** nommé *m1* et un **routeur-firewall** nommé *proxy*. Utilisez « debian-wheezy » comme système pour *m1*, et « guignol » pour *proxy*.
   - La machine *m1* aura comme adresse IP 192.168.40.1.
   - La machine *proxy* aura 2 interfaces réseau eth0 et eth1 (pensez à préciser 2 cartes Ethernet dans la fenêtre de création de la machine) respectivement d'adresses 192.168.40.254 et 195.83.80.254. Pensez également à lui définir 2 consoles.



- Sur le **client** *m1* :
  1) Configurez l'interface réseau et la table de routage

- Sur le **routeur-firewall** *proxy* :
  1) Configurez les deux interfaces réseau et la table de routage correspondante

  2) Activez le routage avec la commande :
     ```
     echo 1 > /proc/sys/net/ipv4/ip_forward
     ```
     supprimer également la redirection de route (ICMP redirect)
     ```
     echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects
     echo 0 > /proc/sys/net/ipv4/conf/eth0/send_redirects
     ```

  3) Activez le camouflage IP :
     ```
     iptables -A POSTROUTING -t nat -j MASQUERADE
     ```

Vérifiez que le camouflage est activé :

```
iptables -t nat -L
```

```
Chain PREROUTING (policy ACCEPT)
target     prot opt     source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt     source               destination
MASQUERADE  all  --      anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt     source               destination
```

4) Autorisez le routage des paquets venant du réseau privé
   `iptables -A FORWARD -s 192.168.40.0/24 -j ACCEPT`
   Vérifiez que le routage des paquets est autorisé :

```
iptables -L
```

```
Chain INPUT (policy ACCEPT)

Chain FORWARD (policy ACCEPT)
target     prot opt     source               destination
ACCEPT     all  --      192.168.40.0/24      anywhere

Chain OUTPUT (policy ACCEPT)
```

Pour vous assurer que la configuration est correcte, depuis la machine *m1* de votre réseau privé :

1) Utilisez la commande `ping` pour atteindre 195.83.80.10.
2) Lancer le navigateur avec la commande :
   `epiphany http://195.83.80.10/whatismyip.php`
   Observez l'adresse IP affichée dans la page et vérifiez qu'elle correspond bien à celle de votre passerelle.

Mise en place de règles de filtrage simples : étape ❷

Pour apprendre à manipuler les règles de filtrage, nous allons commencer par bloquer le `ping` sur l'adresse de bouclage (127.0.0.1) **du routeur-firewall** *proxy*.

1) Ajoutez dans votre réseau privé, une nouvelle machine *m2* d'adresse IP 192.168.40.2. Configurez son interface réseau et sa table de routage.

2) Sur *proxy*, vérifiez que votre interface de bouclage fonctionne correctement (0% des paquets perdus) :
```
ping -c 5 127.0.0.1
```

3) Créez une nouvelle chaîne utilisateur nommée `LOG_DROP` pour à la fois rejeter les paquets et enregistrer dans le journal (fichier `/var/log/messages`) les paquets rejetés.
```
iptables -N LOG_DROP
iptables -A LOG_DROP -j LOG
iptables -A LOG_DROP -j DROP
```

4) Appliquez un filtre sur la chaîne d'entrée INPUT :
```
iptables –A INPUT –p icmp –s 127.0.0.1 –j LOG_DROP
```

5) Vérifiez que votre modification a bien été prise en compte en affichant la chaîne INPUT :
```
iptables –L INPUT
```

6) Connectez-vous sur le 2e terminal puis tapez la commande :
```
tail –f /var/log/messages
```
Dans le premier terminal, essayez de faire à nouveau le `ping` ; la connexion ne doit pas pouvoir aboutir (100% des paquets perdus). Observez les messages qui s'affichent au fur et à mesure dans le terminal dans lequel vous avez lancé la commande `tail`.

7) Supprimez la règle (de numéro 1) sur la chaîne INPUT :
```
iptables –D INPUT 1
```
ou
```
iptables –D INPUT –p icmp –s 127.0.0.1 –j LOG_DROP
```

8) Toujours **sur le routeur-firewall**, trouvez à présent une règle pour interdire le `ping` depuis n'importe quelle machine de votre réseau privé. Une fois que vous avez testé le bon fonctionnement de la règle depuis les deux machines, supprimez-la sur le routeur-firewall.

Mise en place de règles de filtrage plus élaborées : étape ❸

- A présent, votre but est **d'autoriser toutes les connexions sortantes sauf les connexions HTTP** (sur le port 80) depuis votre réseau privé vers le réseau public (Internet).

  1) Editez le fichier `/etc/resolv.conf` de *m1* et *m2*, supprimez toutes les lignes et ajoutez uniquement la ligne :
  `nameserver 195.83.80.3`

  2) Essayez de vous connecter avec un navigateur (epiphany) depuis les postes clients sur `http://www.http2demo.io/`

  3) Pour vérifier l'état des différentes chaînes **sur le routeur-firewall**, tapez la commande :
  `iptables -L`
  Le camouflage doit être activé et la chaîne utilisateur `LOG_DROP` doit être présente (cf étapes précédentes).

```
Chain INPUT (policy ACCEPT)

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  192.168.40.0/24      anywhere

Chain OUTPUT (policy ACCEPT)

Chain LOG_DROP (0 references)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere            LOG level warning
DROP       all  --  anywhere             anywhere
```

  4) Interdisez maintenant en TCP :
  - l'accès aux paquets venant du port 80 (réponses HTTP)
  - l'accès aux paquets allant vers le port 80 (requêtes HTTP)

```
iptables -A FORWARD -p tcp --sport 80 -m state --state ESTABLISHED
-j LOG_DROP
iptables -A FORWARD -p tcp --dport 80 -m state --state NEW,ESTABLISHED
-j LOG_DROP
```

Le résultat `iptables -L` doit être le suivant :

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  192.168.40.0/24      anywhere
LOG_DROP   tcp  --  anywhere             anywhere            tcp spt:http state
ESTABLISHED
LOG_DROP   tcp  --  anywhere             anywhere            tcp dpt:http state
NEW,ESTABLISHED

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain LOG_DROP (2 references)
target     prot opt source               destination
LOG        all  --  anywhere             anywhere            LOG level warning
DROP       all  --  anywhere             anywhere
```

5) A partir de maintenant, il est impossible d'accéder depuis les stations du réseau privé (firewall exclu) aux pages web (par HTTP) dont le serveur fonctionne sur le port 80.
   - Testez le rejet des connexions HTTP sortantes en essayant de vous connecter sur `http://www.http2demo.io/` depuis les postes clients.

6) Observez les traces des paquets rejetés sur le firewall. Vous constaterez que ce sont les réponses et non les requêtes qui sont bloquées. Expliquez pourquoi. Quelle(s) modification(s) devez-vous apporter pour que les requêtes et non les réponses soient bloquées ?

- Il s'agit à présent de résoudre le problème inverse, c'est-à-dire **interdire toutes les connexions sortantes sauf celles du protocole HTTP** (sur le port 80).

Il faut ne pas oublier d'autoriser les connexions vers le service de DNS, sinon la résolution de nom sera impossible.

1) Tout d'abord, supprimez les règles que vous avez définies dans la chaîne FORWARD
```
iptables -F FORWARD
```

2) Modifiez ensuite la politique (en « tout ce qui n'est pas explicitement autorisé est interdit ») pour la chaîne FORWARD (DROP)
```
iptables -P FORWARD DROP
```

3) Il faut ajouter les règles pour accéder au DNS (en UDP et en TCP au cas où les réponses du serveur DNS dépassent 512 octets) :

```
iptables -A FORWARD -p udp –s 0/0 –d 192.168.40.0/24 --sport 53
-j ACCEPT
iptables -A FORWARD -p udp –s 192.168.40.0/24 –d 0/0 --dport 53
-j ACCEPT
iptables -A FORWARD -p tcp –s 0/0 –d 192.168.40.0/24 --sport 53
-j ACCEPT
iptables -A FORWARD -p tcp –s 192.168.40.0/24 –d 0/0 --dport 53
-j ACCEPT
```

et autoriser les connexions HTTP :

```
iptables -A FORWARD -p tcp –s 0/0 –d 192.168.40.0/24 --sport 80
-m state --state ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp –s 192.168.40.0/24 –d 0/0 --dport 80
-m state --state NEW,ESTABLISHED -j ACCEPT
```

4) Assurez-vous qu'il est à présent seulement possible, à partir du réseau privé :
   - d'effectuer des requêtes DNS en UDP et en TCP (en utilisant la commande `nslookup` par exemple : `nslookup miashs-www.u-ga.fr`).
   - d'effectuer des requêtes HTTP (en utilisant un browser web).

- **Ajoutez** sur le routeur-firewall **les règles nécessaires pour** permettre, depuis les clients, la connexion par `ssh` sur la machine « extérieure » miashs-dc.u-ga.fr (pour tester, utilisez la commande : `ssh miashs-dc.u-ga.fr`)