Configuration d’un parefeu Netfilter sous GNU/Linux

Objectifs

- Comprendre le fonctionnement d’un parefeu (firewall) netfilter et d’une DMZ
- Étudier les règles de filtrage et de translation entre réseaux privés et public.
- Mettre en place un routeur pare-feu (firewall) entre deux réseaux aux moyens d’outils GNU/Linux

On attend au minimum des réponses devant chaque point numéroté. Il est également possible et conseillé de rajouter tout commentaire et/ou notes personnelles qui pourront vous servir par la suite.

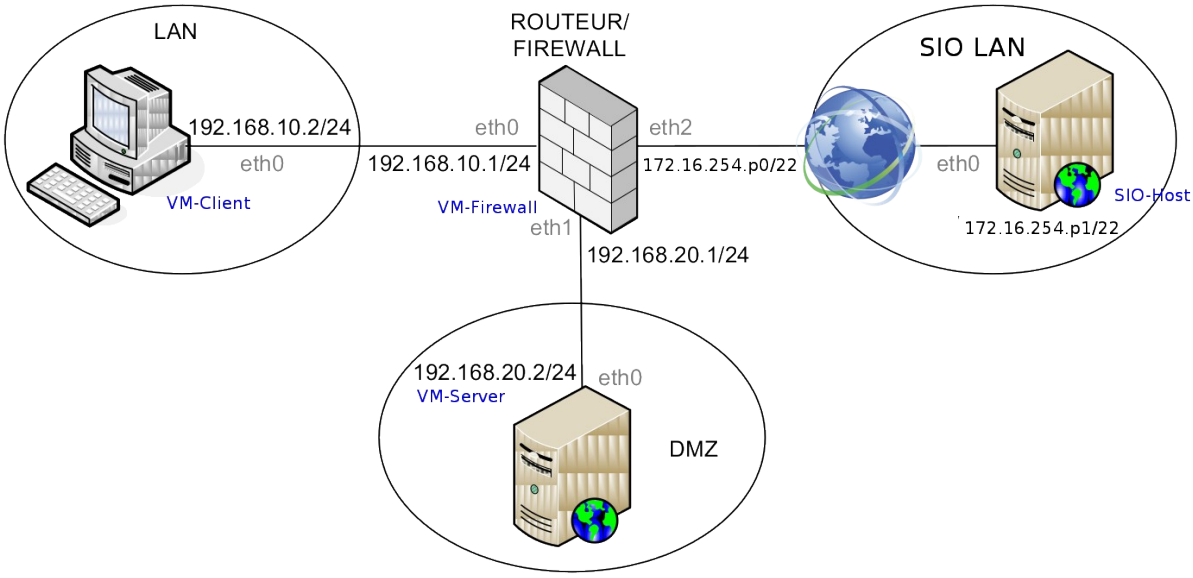
Configuration générale

L’ensemble de la configuration générale sera effectuée à partir de quatre postes :

- Votre poste personnel relié au réseau **SIO LAN**. Ce poste sera nommé par la suite **SIO-Host**. Il servira de client externe à l’infrastructure.
- Une machine virtuelle sous Windows ou GNU/Linux graphique qui servira de poste client sur le réseau privé local **LAN**. Elle sera nommée par la suite **VM-Client**.
- Une machine virtuelle GNU/Linux textuelle qui hébergera un serveur web apache en zone démilitarisée **DMZ**. Elle sera nommée par la suite **VM-Server**.
- Une machine virtuelle GNU/Linux textuelle qui sera configurée pour prendre le rôle de routeur/firewall sous Debian GNU/Linux. On l’appellera **VM-Firewall**.

L’objectif est de configurer un parefeu (firewall) permettant à une entreprise de filtrer les accès vers son réseau privé et de rendre accessible à partir d’Internet un serveur web placé sur une zone neutre de type DMZ. Les adresses du réseau privé et du serveur web seront masquées pour l’extérieur en utilisant le principe du NAT⁄PAT. Le firewall utilisé fonctionnera sous GNU/Linux avec la solution netfilter et les règles de filtrage ***iptables***.
Pour simplifier, le réseau local privé LAN est représenté par la machine VM-Client.
L’Internet est émulé par le réseau SIO LAN et est représenté par la machine SIO-Host et le serveur web sur la DMZ par la machine VM-Server. L’accès entre les différentes machines est géré par le firewall hébergé sur la macine VM-Firewall.

Le schéma suivant présente l’architecture du réseau avec le plan d’adressage :



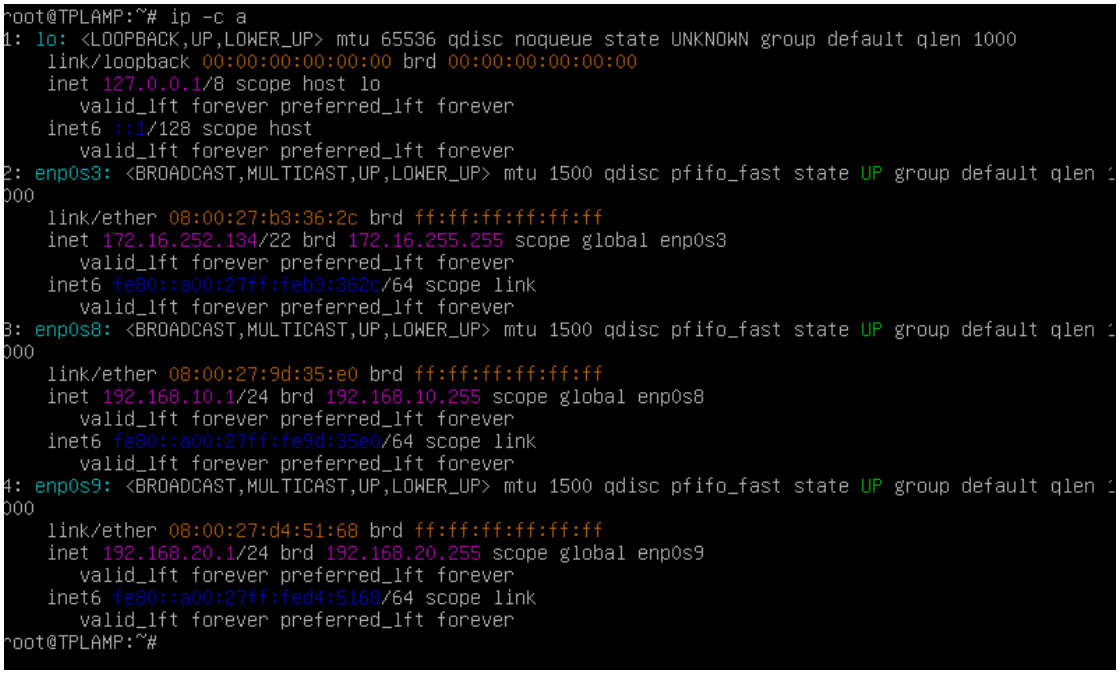
Le firewall doit répondre aux consignes suivantes :

- il autorise l'accès vers un serveur web *d'Internet* (SIO LAN) à partir du LAN ;
- il doit permettre le ping d'une machine vers une machine d'*Internet* (SIO LAN) (message echo request) ;
- il doit accepter en retour la réponse du ping (echo reply) ;
- il ne doit pas autoriser une demande de connexion à partir d'une machine venant d'*Internet* (SIO LAN) ;
- les machines du LAN ne doivent pas être visibles d'*Internet* (SIO LAN) ;
- les machines du LAN doivent pouvoir accéder au serveur web de l'entreprise localisé dans la DMZ ;
- une machine *d'Internet* (SIO LAN) doit pouvoir accéder au serveur web de la DMZ mais l'adresse de ce dernier doit être masquée de l'extérieur.

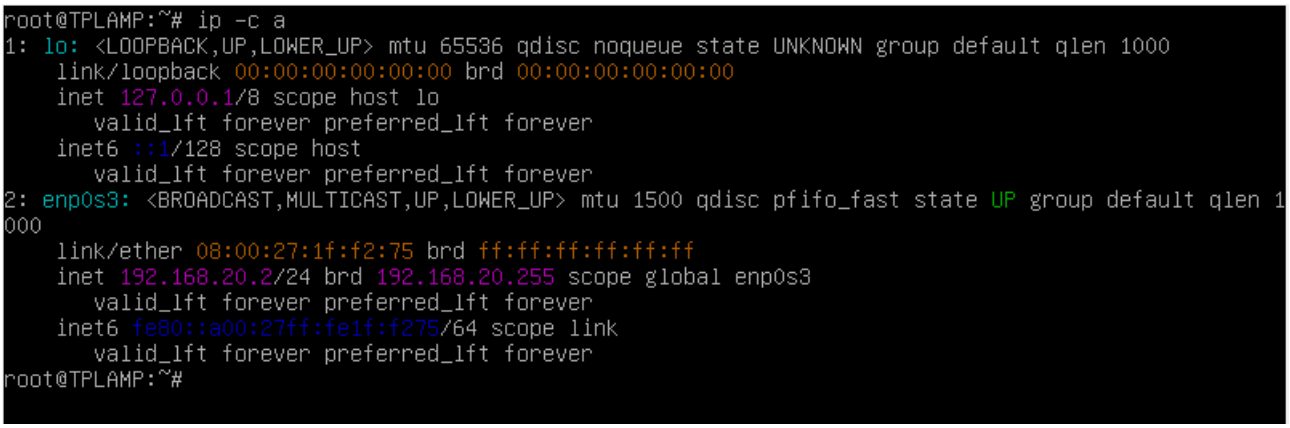
Partie 1 : mise en œuvre du routage sans sécurité

Le but de cette partie est de réaliser l'interconnexion des quatre machines en configurant leurs interfaces Ethernet, en activant le routage IP sur la machine VM-Firewall et en configurant les tables de routage des machines sans contrôler les accès (pas de règle de filtrage particulière).

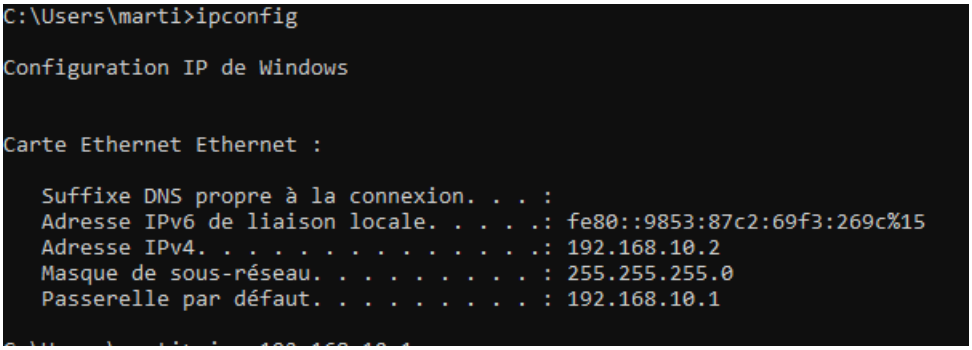
1. Attribuez aux machines les adresses IP suivant le plan d'adressage fourni dans le schéma et avec les masques adéquats.



Config IP routeur/firewall



Config IP serveur WEB



Config IP Windows

On configurera l'ensemble des machines virtuelles avec l'option mode d'accès réseau de VirtualBox réglée sur « Réseau Interne » (il y aura donc deux réseau privés internes) sauf pour l'interface eth2 de la VM-Firewall qui sera en mode « accès par pont »

Les noms des réseaux internes affectés seront les suivants :

- intnetLAN pour les machines VM-Client et VM-Firewall-eth1
- intnetDMZ pour les machines VM-Server et VM-Firewalm-eth2

2. Éditer les connexions nécessaires entre les machines en utilisant l'outil permettant de créer des tables de routages (« ip route », pour la vérification, /etc/network/interface pour la configuration). On configurera donc la passerelle par défaut notamment.

```
root@TPLAMP:~# ip route
default via 172.16.255.254 dev enp0s3 onlink
169.254.0.0/16 dev enp0s3 scope link metric 1000
172.16.252.0/22 dev enp0s3 proto kernel scope link src 172.16.252.134
192.168.10.0/24 dev enp0s8 proto kernel scope link src 192.168.10.1
192.168.20.0/24 dev enp0s9 proto kernel scope link src 192.168.20.1
root@TPLAMP:~#
```

Quelques éléments utiles :

- Pour qu'une machine fasse suivre les paquets d'une interface vers une autre (fonctionnement en mode routeur) il faut activer `ip_forward` au niveau du noyau :

`#sysctl -w net.ipv4.ip_forward=1` (=0 pour désactiver le mode routeur)

- Pour ajouter une entrée dans la table de routage d'u poste : (Cf. **ip route help**)

`ip route add` *`@reseau/nb_bits_reseau`* **`via`** *`@routeur_pour_aller_sur_ce_reseau`*

- Pour enlever une entrée dans la table de routage :

`ip route del` *`@reseau/nb_bits_reseau`* **`via`** *`@routeur_pour_aller_sur_ce_reseau`*

3. Vérifiez à l'aide de la commande ping sur les différents postes que vous pouvez accéder de n'importe quelle machine vers toutes les autres machines. En effet, pour l'instant le filtrage n'est pas activé.

Résultat obtenue	Firewall	DMZ	Client Windows	SIO LAN
Firewall	/	Oui	Oui	Oui
DMZ	Oui	/	Oui	Oui
Client windows	Oui	Oui	/	Oui

Attention également ici, cette manipulation n'est que temporaire
Pour rendre l'ensemble des opérations plus simple à réaliser la fois suivante, le mieux est de les enregistrer dans un fichier de script qui sera exécuté lorsque l'on voudra obtenir la même configration du routeur/firewall à nouveau.
Création d'un fichier de script.

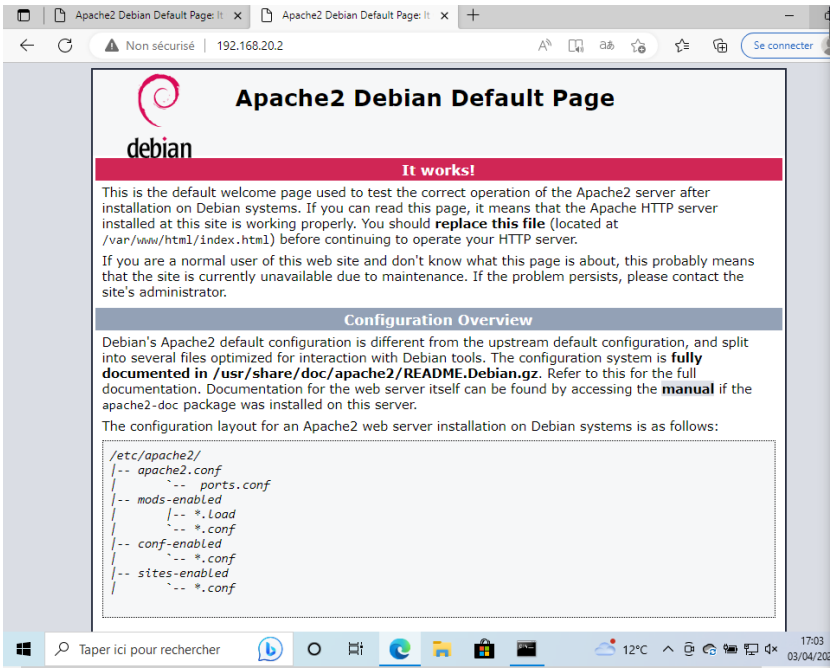
Ce fichier pourra être appelé /etc/router.sh. Il contiendra l'ensemble des routes de notre routeur.
Pour l'exécuter il faudra d'abord le rendre exécutable par la commande « chmod 700 /etc/router.sh »
Pour tester : On tape /etc/router.sh dans un shelle, ce qui lancera l'interprétation des commandes du script.

4. Vérifiez sur la machine de la DMZ que le serveur web est opérationnel (commande `systemctl status apache2.service`). Si ce n'est pas le cas, procédez aux modifications nécessaires.

```
root@TPLAMP:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-04-03 13:45:07 CEST; 36min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 436 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 493 (apache2)
      Tasks: 6 (limit: 1130)
     Memory: 36.2M
        CPU: 265ms
     CGroup: /system.slice/apache2.service
             ├─493 /usr/sbin/apache2 -k start
             ├─516 /usr/sbin/apache2 -k start
             ├─517 /usr/sbin/apache2 -k start
             ├─518 /usr/sbin/apache2 -k start
             ├─519 /usr/sbin/apache2 -k start
             └─520 /usr/sbin/apache2 -k start

avril 03 13:45:07 TPLAMP systemd[1]: Starting The Apache HTTP Server...
avril 03 13:45:07 TPLAMP apachectl[460]: AH00558: apache2: Could not reliably determine the server'
avril 03 13:45:07 TPLAMP systemd[1]: Started The Apache HTTP Server.
lines 1-20/20 (END)
```

5. À partir de la machine sur le réseau LAN, ouvrez un navigateur et connectez-vous au serveur web pour vérifier son fonctionnement.



Partie 2 : Configuration par défaut du parefeu

La commande `iptables` sous GNU/Linux permet de définir les règles de filtrage des flux en entrée ou en sortie sur le routeur/firewall (beaucoup de tutoriaux sur iptables existent sur Internet, cherchez). Cette commande utilise trois chaînes :

- INPUT pour les paquets reçus sur une interface ;
- OUTPUT pour les paquets qui sont générés localement et qui sortent d'une interface ;
- FORWARD pour les paquets qui entrent par une interface et qui sortent par une autre.

Pour chaque chaîne, trois traitements sont possibles pour les paquets : ACCEPT, REJECT ou DROP.

Quelques commandes de base :

- Pour afficher l'état courant du firewall : **`iptables -L`**
- Pour supprimer toutes les règles du firewall (sauf les règles par défaut) : **`iptables -F :`**
- Pour définir la politique par défaut : **`iptables -P`** ***`regle -i interface option`***

Exemple pour interdire par défaut tout flux en entrée sur l'interface Ethernet numéro 0 (eth0) :

```
iptables -P INPUT -i eth0 DROP
```

- Pour ajouter une nouvelle règle au firewall :

iptables -A *regle* -i *interface* -s *@reseau/préfixe* -d *@reseau/préfixe* -p *protocole* -j *option*

Parefeu avec état :
Parefeu sans état :

6. Réalisez une configuration par défaut du firewall qui rejette tout flux dans les chaînes INPUT, OUTPUT et FORWARD (commande `iptables -P`). Ceci constitue la politique de sécurité généralement mise en œuvre qui consiste à tout bloquer par défaut puis ensuite à n'autoriser que le trafic légitime. Consignez ici les étapes nécessaires.

Faire un iptables -L pour vérifier la configuration :

```
root@Routeurfirewall:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@Routeurfirewall:~#
```

Ensuite on fais un iptables -F afin d'effacer toutes les règles

Puis enfin on effectue la commande iptables -P FORWARD DROP (changer forward pour input ou output) et enfin vérifier la configuration avec un nouveau iptables -L :

```
root@Routeurfirewall:~# iptables -P FORWARD  DROP
root@Routeurfirewall:~# iptables -P INPUT  DROP
root@Routeurfirewall:~# iptables -P OUTPUT  DROP
root@Routeurfirewall:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
root@Routeurfirewall:~# _
```

7. *Vous pouvez utiliser un fichier script pour éviter d'avoir à saisir plusieurs fois les commandes. Donnez les commandes qui permettent la création de ce script ainsi que les instructions pour pouvoir ensuite s'en servir (on nommera ce fichier firewall.sh).*

Fichier .service :

```
 GNU nano 5.4                    /lib/systemd//system/
[Service]
Description=Demarage routeur/firewall
Type=oneshot
RemainAfterExit=yes
ExecStart=/bin/bash /etc/init.d/firewall.sh

[Install]
WantedBy=multi-user.target
```

Fichier .sh :

```
#!/bin/bash
echo "Execution Script Parefeu Personnel"
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

Il faut maintenant faire en sorte que le fichier s'exécute à chaque démarrage de la machine :

- systemctl enable monscript.service
- systemctl stop monscript.service
- systemctl start monscript.service
- systemctl daemon-reload

puis reboot pour vérifier que le srcipt se lance bien.

8. *Vérifiez maintenant à l'aide de pings que les trois machines (VM-Client , SIO-Host et VM-Server) ne peuvent plus communiquer.*

Sur mes 3 machines les pings sont impossible ce qui veut dire que mon pare-feu à bien bloqué toutes les entrées et les sorties.

```
root@ServeurWEB:~# ping 172.16.252.254
PING 172.16.252.254 (172.16.252.254) 56(84) bytes of data.
```

Partie 3 : autorisation du trafic ICMP (ping)

Le but de cette partie est de rajouter les règles sur le firewall pour autoriser les pings du LAN vers la DMZ.

9. *Ajoutez les règles de filtrage permettant d'accepter sur l'interface du firewall coté LAN un ping venant du LAN. Donnez les règles ajoutées.*

Commande input : iptables -A INPUT -i enp0s8 -s 192.168.10.2/24 -d 192.18.10.1/24 -p ICMP -j ACCEPT

Commande output : -A OUTPUT -o enp0s8 -s 192.168.10.2/24 -d 192.18.10.1/24 -p ICMP -j ACCEPT

10. Exécutez votre script. Faites un ping à partir de VM-Client vers l'interface 192.168.10.1 et vérifiez que cela fonctionne.

```
C:\Users\marti>ping 192.168.10.1

Envoi d'une requête 'Ping'  192.168.10.1 avec 32 octets de données :
Réponse de 192.168.10.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.10.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.10.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.10.1 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.10.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
```

11. *De la même façon ajoutez et testez les filtres pour autoriser les pings entre la DMZ et le firewall.*

```
root@Routeurfirewall:~# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=128 time=0.713 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=128 time=1.30 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=128 time=1.57 ms
^C
--- 192.168.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2029ms
rtt min/avg/max/mdev = 0.713/1.193/1.566/0.356 ms
root@Routeurfirewall:~# _
```

12. Vérifiez et testez que les pings sont désormais possibles entre les deux zones. Si ce n'est pas le cas, procédez aux modifications nécessaires et testez de nouveau.

Commande pour activer les pings entre la DMZ et le client :

iptables -A FORWARD -s 192.168.20.2/24 -i enp0s9 -d 192.168.10.2/24 -o enp0s8 -p ICMP -j ACCEPT

```
root@ServeurWEB:~# ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=127 time=1.26 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=127 time=0.894 ms
^C
--- 192.168.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.894/1.077/1.261/0.183 ms
root@ServeurWEB:~# _
```

```
C:\Users\marti>ping 192.168.20.2

Envoi d'une requête 'Ping'  192.168.20.2 avec 32 octets de données :
Réponse de 192.168.20.2 : octets=32 temps<1ms TTL=63
Réponse de 192.168.20.2 : octets=32 temps=2 ms TTL=63
Réponse de 192.168.20.2 : octets=32 temps=2 ms TTL=63
Réponse de 192.168.20.2 : octets=32 temps=2 ms TTL=63

Statistiques Ping pour 192.168.20.2:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 2ms, Moyenne = 1ms
```

Partie 4 : Translation d'adresses

Le but de cette partie est de réaliser une translation pour toutes les adresses venant du LAN et allant vers la DMZ.
Dans le programme `iptables`, la table « `nat` » est utilisée pour les translations d'adresses. Comme pour le filtrage, cette table utilise trois chaînes :

- PREROUTING pour faire du DNAT (Destination NAT), la translation est réalisée sur l'adresse de destination avant le processus de routage. Exemple : pour un paquet entrant vers un serveur web interne et masqué, le routeur va remplacer sa propre adresse IP par l'adresse du serveur web.
- POSTROUTING utilisée à la sortie du routeur pour faire du SNAT (Source NAT), l'adresse source est masquée après le processus de routage. Exemple : un ordinateur local veut sortir sur Internet, le routeur va remplacer l'adresse IP du paquet émis en local par sa propre adresse.
- OUTPUT pour les paquets qui sont générés localement et qui sortent d'une interface.

Pour les deux premières chaînes, le traitement permettant de faire du masquage est noté MASQUERADE.

Exemple

```
iptables -t nat -A POSTROUTING -s 10.2.0.0/16 -d 10.3.0.0/16 -o eth2
-j MASQUERADE
```

Cette commande ajoute une règle dans la table de translation d'adresses `nat` du routeur qui opère après la décision de routage (postrouting) et qui masque (`masquerade`) le trafic provenant du réseau 10.2.0.0 et à destination du réseau 10.3.0.0.
Ce dernier voit le trafic sortant de l'interface eth2 comme provenant uniquement du routeur.

13. *Ajoutez une règle sur le firewall permettant de faire de la translation d'adresse entre le PC LAN et la DMZ. Donnez la règle ajoutée et justifiez les options choisies (chaîne utilisée, prerouting ou post routing, politique masquerade…).*

    Éxécutez votre script.

Installer l'utilitaire *`tcpdump`* sur la machine VM-Server

14. *À quoi sert cet utilitaire ?*

15. *Donnez un logiciel équivalent que vous avez déjà utilisé et qui comporte une interface graphique.*

16. Sur VM-Server, exécutez la commande `tcpdump –i eth0` pour réaliser une capture de trames. Sur VM-Client, tapez la commande `ping -c 4 192.168.20.2` (on remplacera `eth1` par l'interface de la VM réellement utilisée dans votre configuration).

17. Donnez la séquence des trames obtenue sur VM-Server avant et après la mise en place de la translation. Interprétez les résultats obtenus.

On souhaite maintenant permettre l'accès au serveur web de la DMZ pour les machines du LAN.

18. *Proposez les règles de filtrage pour la chaîne FORWARD permettant une connexion du LAN à destination du serveur web, sur son port d'écoute.*

19. Testez à l'aide d'un navigateur sur VM-Client que l'accès au serveur web fonctionne.

Partie 5 : filtrage entre LAN et SIO LAN

Les machines du LAN ne doivent pas être directement visibles de l'Internet.

20. Ajoutez les règles de filtrage pour accepter un ping et un accès à un serveur web situé sur Internet. On réalisera une translation d'adresse pour toute machine ayant comme adresse source le réseau LAN et comme réseau destination le réseau INTERNET (SIO LAN). Vérifiez les accès et la translation.

Partie 6 : filtrage entre DMZ et SIO LAN

Le serveur web de la DMZ doit être accessible à partir de la machine SIO-Host mais pas directement avec l'adresse 192.168.20.2.
À partir de la machine SIO-Host, seule une connexion avec l'URL suivante devrait fonctionner : *http://172.16.254.p0*.

Pour cela, il faut mettre en place une translation d'adresse du réseau DMZ vers le réseau SIO LAN et un forwarding de port pour que toute connexion HTTP venant de SIO LAN vers la machine *172.16.254.p0* soit redirigée vers la machine VM-Server, sur le port d'écoute du serveur web de la DMZ.

21. *Ajoutez les règles de filtrage `iptables` nécessaires pour réaliser la translation d'adresse et le forwarding de port (vous justifierez les options choisies : prerouting ou post routing, SNAT ou DNAT…).*

22. Testez la configuration sur la machine SIO-Host en utilisant un navigateur avec l'url suivante : *http://172.16.254.p0*.

23. *Analysez avec `tcpdump` les trames obtenues lors de la connexion http précédente* sur le pc VM-Server.

Annexes

- Iptable par l'exemple − Léa Linux − http://lea-linux.org/documentations/Iptables
- Introduction à Netfilter et iptables − https://connect.ed-diamond.com/GNU-Linux-Magazine/GLMFHS-041/Introduction-a-Netfilter-et-iptables
- Parefeu Netfilter et iptables − documentation fedora − https://doc.fedora-fr.org/wiki/Parefeu_-_firewall_-_netfilter_-_iptables
- iptables − article Wikipedia
- Netfilter − article Wikipedia
- nftables − article Wikipedia
- Exemple de fichier de script permettant la configuration d'un parefeu Netfilter avec iptables − http://www.canonne.net/linux/iptables/firewall.sh.php?print=1
- Présentation de NFTables − IT-Connect
- Quick reference-nftables in 10 minutes − https://wiki.nftables.org/wiki-nftables/index.php/Quick_reference-nftables_in_10_minutes