Réseau

- Firewall / Pare-feu

Firewall / Pare-feu

Installation d'un pare-feu

Installation de iptables

```
apt install iptables
```

Création du process et des règles

Je créer un fichier firewall qui va être exécuter au démarrage et gérer par systemctl.

Son but et d'indiquer quoi faire quand on start le process et quand on le stop et la gestion du status.

```
nano /etc/init.d/firewall
```

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:          Firewall
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Firewall
# Description:       Firewall
### END INIT INFO

IPT=/sbin/iptables
IP6T=/sbin/ip6tables

do_start() {
    # Efface toutes les regles en cours. -F toutes. -X utilisateurs
    $IPT -t filter -F
```

```
    $IPT -t filter -X
    $IPT -t nat -F
    $IPT -t nat -X
    $IPT -t mangle -F
    $IPT -t mangle -X
    $IP6T -t filter -F
    $IP6T -t filter -X
    $IP6T -t mangle -F
    $IP6T -t mangle -X
    # strategie (-P) par defaut : bloc tout l'entrant le forward et autorise le sortant
    $IPT -t filter -P INPUT DROP
    $IPT -t filter -P FORWARD DROP
    $IPT -t filter -P OUTPUT ACCEPT
    $IP6T -t filter -P INPUT DROP
    $IP6T -t filter -P FORWARD DROP
    $IP6T -t filter -P OUTPUT ACCEPT
    # Loopback
    $IPT -t filter -A INPUT -i lo -j ACCEPT
    $IPT -t filter -A OUTPUT -o lo -j ACCEPT
    $IP6T -t filter -A INPUT -i lo -j ACCEPT
    $IP6T -t filter -A OUTPUT -o lo -j ACCEPT
    # Permettre a une connexion ouverte de recevoir du trafic en entree
    $IPT -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
    $IP6T -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

    echo "firewall started [OK]"
}
# fonction qui arrete le firewall
do_stop() {
    # Efface toutes les regles
    $IPT -t filter -F
    $IPT -t filter -X
    $IPT -t nat -F
    $IPT -t nat -X
    $IPT -t mangle -F
    $IPT -t mangle -X
    $IP6T -t filter -F
    $IP6T -t filter -X
    $IP6T -t mangle -F
    $IP6T -t mangle -X
```

```
    # remet la strategie
    $IPT -t filter -P INPUT ACCEPT
    $IPT -t filter -P OUTPUT ACCEPT
    $IPT -t filter -P FORWARD ACCEPT
    $IP6T -t filter -P INPUT ACCEPT
    $IP6T -t filter -P OUTPUT ACCEPT
    $IP6T -t filter -P FORWARD ACCEPT
    #
    echo "firewall stopped [OK]"
}
# fonction status firewall
do_status() {
    # affiche les regles en cours
    clear
    echo Status IPV4
    echo -----------------------------------------------
    $IPT -L -n -v
    echo
    echo -----------------------------------------------
    echo
    echo status IPV6
    echo -----------------------------------------------
    $IP6T -L -n -v
    echo
}
case "$1" in
    start)
        do_start
        exit 0
    ;;
    stop)
        do_stop
        exit 0
    ;;
    restart)
        do_stop
        do_start
        exit 0
    ;;
    status)
```

```
        do_status
        exit 0
    ;;
    *)
        echo "Usage: /etc/init.d/firewall {start|stop|restart|status}"
        exit 1
    ;;
esac
```

Nous avons ici la configuration la plus bloquante (trafic sortant uniquement): Uniquement le trafic local et les connexion établie vers l'extérieur en IPv4 et IPv6.

Pour ajouter des autorisation, par exemple l'ICMP (ping), dans `do_start()` ajouter:

```
# ICMP
    $IPT -t filter -A INPUT -p icmp -j ACCEPT
    $IP6T -t filter -A INPUT -p ipv6-icmp -j ACCEPT
```

Votre `do_start()` ressemble à ça:

```
do_start() {
    # Efface toutes les regles en cours. -F toutes. -X utilisateurs
    $IPT -t filter -F
    $IPT -t filter -X
    $IPT -t nat -F
    $IPT -t nat -X
    $IPT -t mangle -F
    $IPT -t mangle -X
    $IP6T -t filter -F
    $IP6T -t filter -X
    $IP6T -t mangle -F
    $IP6T -t mangle -X
    # strategie (-P) par defaut : bloc tout l'entrant le forward et autorise le sortant
    $IPT -t filter -P INPUT DROP
    $IPT -t filter -P FORWARD DROP
    $IPT -t filter -P OUTPUT ACCEPT
    $IP6T -t filter -P INPUT DROP
    $IP6T -t filter -P FORWARD DROP
    $IP6T -t filter -P OUTPUT ACCEPT
    # Loopback
    $IPT -t filter -A INPUT -i lo -j ACCEPT
```

```
    $IPT -t filter -A OUTPUT -o lo -j ACCEPT
    $IP6T -t filter -A INPUT -i lo -j ACCEPT
    $IP6T -t filter -A OUTPUT -o lo -j ACCEPT
    # Permettre a une connexion ouverte de recevoir du trafic en entree
    $IPT -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
    $IP6T -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

 # ICMP
    $IPT -t filter -A INPUT -p icmp -j ACCEPT
    $IP6T -t filter -A INPUT -p ipv6-icmp -j ACCEPT

    echo "firewall started [OK]"
}
```

Voici une liste de règles souvent utile

```
    # ICMP
    $IPT -t filter -A INPUT -p icmp -j ACCEPT
    $IP6T -t filter -A INPUT -p ipv6-icmp -j ACCEPT

    # DNS:53
    $IPT -t filter -A INPUT -p tcp --dport 53 -j ACCEPT
    $IPT -t filter -A INPUT -p udp --dport 53 -j ACCEPT
    $IP6T -t filter -A INPUT -p tcp --dport 53 -j ACCEPT
    $IP6T -t filter -A INPUT -p udp --dport 53 -j ACCEPT

    #FTP:20+21 - FTP pasv: 10 090 - 10 100
    $IPT -t filter -A INPUT -p tcp --dport 20 -j ACCEPT
    $IP6T -t filter -A INPUT -p tcp --dport 20 -j ACCEPT
    $IPT -t filter -A INPUT -p tcp --dport 21 -j ACCEPT
    $IP6T -t filter -A INPUT -p tcp --dport 21 -j ACCEPT
    $IPT -t filter -A INPUT -p tcp --dport 10090:10100 -j ACCEPT
    $IP6T -t filter -A INPUT -p tcp --dport 10090:10100 -j ACCEPT

    # SSH:22
    # ATTENTION, indiques bien ton port personnalisé si tu l'as changé
    $IPT -t filter -A INPUT -p tcp --dport 22 -j ACCEPT
    $IP6T -t filter -A INPUT -p tcp --dport 22 -j ACCEPT

    # HTTP:80 (Serveur Web)
```

```
    $IPT -t filter -A INPUT -p tcp --dport 80 -j ACCEPT
    $IP6T -t filter -A INPUT -p tcp --dport 80 -j ACCEPT

    # HTTPS:443 (Serveur Web)
    $IPT -t filter -A INPUT -p tcp --dport 443 -j ACCEPT
    $IP6T -t filter -A INPUT -p tcp --dport 443 -j ACCEPT

    # SNMP:161 (Monitoring)
    $IPT -t filter -A INPUT -p udp --dport 161 -j ACCEPT
    $IP6T -t filter -A INPUT -p udp --dport 161 -j ACCEPT

    # SMTP:25 (Mail)
□$IPT -t filter -A INPUT -p tcp --dport 25 -j ACCEPT
□$IP6T -t filter -A INPUT -p tcp --dport 25 -j ACCEPT

    # POP3:110 (Mail)
□$IPT -t filter -A INPUT -p tcp --dport 110 -j ACCEPT
□$IP6T -t filter -A INPUT -p tcp --dport 110 -j ACCEPT

□# IMAP:143 (Mail)
□$IPT -t filter -A INPUT -p tcp --dport 143 -j ACCEPT
□$IP6T -t filter -A INPUT -p tcp --dport 143 -j ACCEPT
```

Il faut maintenant rendre exécutable de fichier

```
chmod +x /etc/init.d/firewall
```

> Les prochaines étapes vont rendre actif le pare-feu. Si vous êtes en SSH et que une règle n'autorise pas le SSH vous risquer de vous auto bloqué a la prochaine connexion. Assurez-vous d'avoir un accès directe à la machine pour gérer le cas ou vous seriez dans l'incapacité de vous reconnecter.

Premier démarrage du pare-feu

```
/etc/init.d/firewall start
```

Gestion par systemctl

```
update-rc.d firewall defaults
systemctl enable firewall
systemctl start firewall
systemctl status firewall
```

Cela devrais vous indiquer quelque chose du genre:

```
● firewall.service - LSB: Firewall
   Loaded: loaded (/etc/init.d/firewall; generated)
   Active: active (exited) since *** ****-**-** **:**:** ****; * ago
     Docs: man:systemd-sysv-generator(8)
  Process: ***** ExecStart=/etc/init.d/firewall start (code=exited, status=0/SUCCESS)

***. ** **:**:** monitor systemd[1]: Starting LSB: Firewall...
***. ** **:**:** monitor firewall[30462]: firewall started [OK]
***. ** **:**:** monitor systemd[1]: Started LSB: Firewall.
```