|  |  |  |
| --- | --- | --- |
| iptables  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)    http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  iptables (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png |  | Pare-feu sous Linux  Configuration du pare-feu  Dans ce document, nous allons configurer notre pare-feu de la manière suivante   * On bloque tout le trafic entrant par défaut. * On autorise au cas par cas:  le trafic appartenant ou lié à des connexions déjà établies  et  le trafic à destination des serveurs (web, ssh, etc.) que nous souhaitons mettre à disposition.   Pour voir les règles *exitantes* …  **$ sudo su**  **$ iptables -L**  En entrant la commande iptables -L une liste de vos règles actuelles est affichée.  Si vous (ou un logiciel) n'avez encore jamais touché à *iptables*, les chaines sont vides, et vous devriez voir ce qui suit…  **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**  Pour l'instant, tout passe dans toutes les directions (*policy ACCEPT*).  Pour cette configuration basique, seul le trafic entrant (chaine input) nous intéresse.  Par défaut, iptables -L n'affiche que la table « filter ».  Pour consulter les autres tables, vous devez ajouter l'option -t suivie de « nat », « mangle » ou « raw ».  Pour la configuration d'un pare-feu la table "filter" est toutefois la seule nécessaire.  Attention… Si vous avez modifié la règle par défaut pour le blocage (iptables -P INPUT DROP voir plus bas) et que vous entrez iptables -F vous bloquerez tous les accès … y compris celui en cours.  Ceci est particulièrement problématique sur un hôte sur laquelle vous accédez à distance.  Si vous avez déjà modifié la configuration et que vous voulez la réinitialiser, entrez…  **$ iptables –F**  **$ iptables –X**  **Autoriser le trafic entrant d'une connexion déjà établie**  Pour permettre à une connexion déjà ouverte de recevoir du trafic ...  **$** **iptables -A INPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT**  Le module *conntrack* permet de paramétrer dans quel état se trouve une connexion.  Pour permettre le trafic entrant sur un port spécifique…  Pour permettre le trafic entrant sur le port 22 (traditionnellement utilisé par SSH, vous devrez indiquer à *iptables* tout le trafic TCP sur le port 22 de votre interface réseau.  **$** **iptables -A INPUT -p tcp -i eth0 --dport ssh -j ACCEPT**  Cette commande ajoute une règle (-A) à la chaine contrôlant le trafic entrant INPUT, pour autoriser le trafic (-j ACCEPT), vers l'interface (-i) eth0 et à destination du port (--dport) SSH (on aurait pu mettre 22).  Maintenant vous pouvez vérifier vos règles iptables :  **$ iptables –L**  **Chain INPUT (policy ACCEPT)**  **target prot opt source destination**  **ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED**  **ACCEPT tcp -- anywhere anywhere tcp dpt:ssh**  Maintenant, acceptons tout le trafic web (www) entrant…  **$** **iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT**  En regardant nos règles, nous avons…  **$ iptables -L**  **Chain INPUT (policy ACCEPT)**  **target prot opt source destination**  **ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED**  **ACCEPT tcp -- anywhere anywhere tcp dpt:ssh**  **ACCEPT tcp -- anywhere anywhere tcp dpt:www**  Nous avons exceptionnellement autorisé le trafic tcp pour ssh et les ports web mais, comme nous n'avons rien bloqué, tout le trafic passe quand même.  **Bloquer le trafic**  Maintenant que nous avons fini avec les autorisations, il faut maintenant bloquer le reste. Nous allons en fait modifier la politique par défaut (policy) de la chaine INPUT.  Cette décision (DROP) s'applique lorsqu'aucune règle n'a été appliquée à un paquet.  Donc, si la tentative de connexion n'est permise par aucune des règles précédentes, elle sera rejetée.  **$** **iptables -P INPUT DROP # Attention… À ne pas utiliser sur un serveur distant !**  **$** **iptables -L**  **Chain INPUT (policy DROP)**  **target prot opt source destination**  **ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED**  **ACCEPT tcp -- anywhere anywhere tcp dpt:ssh**  **ACCEPT tcp -- anywhere anywhere tcp dpt:www**  Un autre moyen de procéder est l'ajout en fin de chaine d'une règle supprimant les paquets (les paquets autorisés par les règles précédentes n'atteindraient pas celle-ci), à partir de la commande iptables -A INPUT -j DROP. Il faudrait alors faire attention à la position des futures règles.  **Autoriser le trafic local**  Un p'tit problème de notre configuration est que même l'interface locale (*loopback*) est bloquée.  Nous pourrions avoir écrit les règles de rejet seulement pour *eth0* en spécifiant -i eth0 mais nous pouvons aussi ajouter une règle pour *loopback*.  Par exemple, nous pourrions l'insérer en seconde position…  **$** **iptables -I INPUT 2 -i lo -j ACCEPT**  Pour lister les règles plus en détail…  **$** **iptables -L -v -n**  **Autoriser les requêtes ICMP (*ping*)**  Il peut être utile de valider les réponses aux requêtes *ping*, ne serait-ce que pour s'assurer que le poste est toujours en activité.  **$** **iptables -A OUTPUT -p icmp -m conntrack --ctstate NEW,ESTABLISHED,RELATED -j ACCEPT**  On autorise les *ping*  **$** **iptables -A INPUT -p icmp -j ACCEPT**  **Supprimer une règle**  Si vous vous êtes trompé dans la création d'une règle et que cela vous bloque une connexion, vous pouvez supprimer une seule entrée plutôt que de tout réinitialiser.  Tout d'abord vous listez l'ensemble de vos régles avec l'affichage des lignes :  **$** **iptables -L --line-numbers**  Voici un exemple de ce que cela peut retourner …  **Chain INPUT (policy DROP)**  **num target prot opt source destination**  **1 DROP icmp -- anywhere anywhere**  **2 ACCEPT tcp -- anywhere anywhere tcp dpt:ssh**  **3 ACCEPT tcp -- anywhere anywhere tcp dpt:www**  **4 ACCEPT tcp -- anywhere anywhere tcp dpt:webmin**  **Chain FORWARD (policy ACCEPT)**  **num target prot opt source destination**  **Chain OUTPUT (policy ACCEPT)**  **num target prot opt source destination**  **1 ACCEPT tcp -- anywhere anywhere tcp spt:www**  **2 ACCEPT tcp -- anywhere anywhere tcp spt:12345**  Je souhaite supprimer la ligne 2 de la chaîne OUTPUT  **$** **iptables -D chaine numéro\_de\_ligne**  **$** **iptables -D OUTPUT 2**  **Sauvegarder vos règles**  **$** **iptables-save -c**  **Appliquer les règles au démarrage**  Vous avez testé vos règles alors il reste à les appliquer au démarrage.  **$** **service iptables-persistent**  Note Si le service n’est pas installé il faut le faire à l’aide de la commande apt-get install iptables-persistent.  Le service *iptables-persistent* prend les arguments…   * save pour sauvegarder les règles, * flush pour vider toutes les règles  et * reload pour les recharger depuis les fichiers précités.   Annexe A Script iptables  Ce script est un exemple, il est à adapter selon ses besoins.  Il peut toutefois être utilisé pour une utilisation courante, il offre une plutôt bonne "protection".  Attention…  iptables -A INPUT -m pkttype –pkt-type broadcast -j DROP peut poser problème au DHCP "le broadcast : pour trouver et dialoguer avec un serveur DHCP, la machine va simplement émettre un paquet spécial de broadcast (broadcast sur 255.255.255.255 avec d'autres informations comme le type de requête, les ports de connexion…) sur le réseau local.  Lorsque le serveur DHCP recevra le paquet de broadcast, il renverra un autre paquet de broadcast (n'oubliez pas que le client n'a pas forcement son adresse IP et que donc il n'est pas joignable directement) contenant toutes les informations requises pour le client" cette commande peut donc empêcher le pc de recevoir son ip par le DHCP  Iptables.sh  **#!/bin/bash**  **IPTS=iptables**  **# Vidage des tables et des règles personnelles**  **$IPTS -t filter -F**  **$IPTS -t filter -X**  **# Interdire toutes connexions entrantes et sortantes**  **$IPTS -t filter -P INPUT DROP**  **$IPTS -t filter -P FORWARD DROP**  **$IPTS -t filter -P OUTPUT DROP**  **# Traitement interface local**  **$IPTS -A INPUT -i lo -j ACCEPT**  **$IPTS -A OUTPUT -o lo -j ACCEPT**  **# Ne pas casser les connexions établies**  **$IPTS -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT**  **$IPTS -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT**  **# Authoriser HTTP et HTTPS**  **$IPTS -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT**  **$IPTS -t filter -A OUTPUT -p tcp --dport 443 -j ACCEPT**  **$IPTS -t filter -A INPUT -p tcp --dport 80 -j ACCEPT**  **$IPTS -t filter -A INPUT -p tcp --dport 443 -j ACCEPT**  **# On rejette les balayages (scans) XMAS et NULL.**  **$IPTS -A INPUT -p tcp --tcp-flags FIN,URG,PSH FIN,URG,PSH -j DROP**  **$IPTS -A INPUT -p tcp --tcp-flags ALL ALL -j DROP**  **$IPTS -A INPUT -p tcp --tcp-flags ALL NONE -j DROP**  **$IPTS -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP**  **# Courriels SMTP POP3S (Modifier les ports en fonction de ses besoin)**  **$IPTS -t filter -A INPUT -p tcp --dport 465 -j ACCEPT**  **$IPTS -t filter -A OUTPUT -p tcp --dport 465 -j ACCEPT**  **$IPTS -t filter -A INPUT -p tcp --dport 993 -j ACCEPT**  **$IPTS -t filter -A OUTPUT -p tcp --dport 993 -j ACCEPT**  **# Rejetter silencieusement tous les paquets de diffusion générale (broadcast)**  **# Attention: cfr. message ci-dessus)**  **$IPTS -A INPUT -m pkttype --pkt-type broadcast -j DROP**  **# On journalise les paquets en entrée.**  **$IPTS -A INPUT -j LOG**  **# On journalise les paquets forward.**  **$IPTS -A FORWARD -j LOG**  **exit 0**  **Sources**  Adapté du texte original <http://doc.ubuntu-fr.org/iptables>  Basé sur https  <http://wiki.ubuntu.com/IptablesHowTo>  Autres références  <http://www.inetdoc.net/guides/iptables-tutorial/howtoread.html> |
|  |  |  |