**Configuration et Utilisation Rapide d’IPTables sur Ubuntu**

Le pare-feu **Iptables** [Linux](https://www.hostinger.fr/tutoriels/chmod-chown-linux) est utilisé pour surveiller le trafic entrant et sortant vers un serveur et le filtrer en fonction des règles définies par l’utilisateur, afin d’empêcher toute personne d’accéder au système. En utilisant Iptables, vous pouvez définir des règles qui n’autoriseront que le trafic sélectionné sur votre serveur. Dans ce didacticiel Iptables, vous apprendrez à sécuriser votre application Web à l’aide d’Iptables.

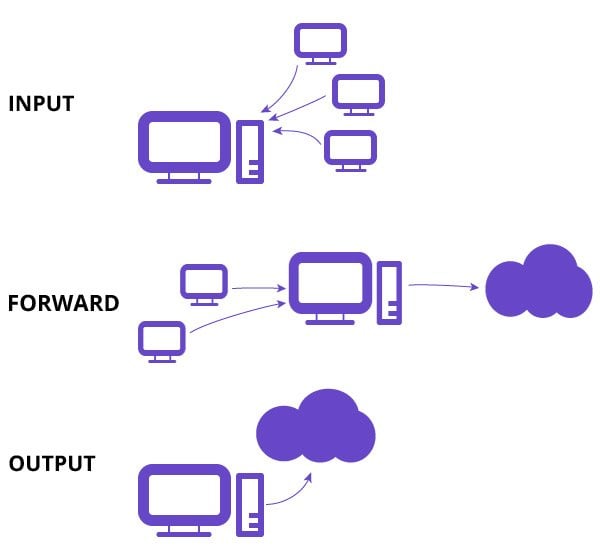
**Les bases d’Iptables**

Toutes les données sont envoyées sous forme de paquets sur Internet. Le noyau Linux fournit une interface pour filtrer les paquets de trafic entrants et sortants à l’aide de tableaux de filtres de paquets. Iptables est une application de ligne de commande et un pare-feu Linux que vous pouvez utiliser pour configurer, maintenir et inspecter ces tableaux. Plusieurs tableaux peuvent être définis. Chaque tableau peut contenir plusieurs chaînes. Une chaîne n’est qu’un ensemble de règles. Chaque règle définit ce qu’il faut faire avec le paquet, s’il correspond à ce paquet. Lorsque le paquet correspond, il lui est attribué une **CIBLE**. Une cible peut être une autre chaîne, pour correspondre à l’une ou l’autre des valeurs spéciales suivantes :

* **ACCEPT** : Cela signifie que le paquet sera autorisé à passer.
* **DROP** : Cela signifie que le paquet ne sera pas autorisé à passer.
* **RETURN** : Cela signifie ignorer la chaîne actuelle et revenir à la règle suivante de la chaîne dans laquelle elle a été appelée.

Pour les besoins de ce didacticiel Iptables, nous allons travailler avec l’un des tableaux par défaut appelés **filtre**. Le tableau des filtres comporte trois chaînes (ensembles de règles).

* **INPUT** – Cette chaîne est utilisée pour contrôler les paquets entrants sur le serveur. **Vous pouvez bloquer/autoriser les connexions en fonction du port, du protocole ou de l’adresse IP source.**
* **FORWARD** – Cette chaîne est utilisée pour filtrer les paquets qui entrent sur le serveur mais doivent être transférés ailleurs.
* **OUTPUT** – Cette chaîne est utilisée pour filtrer les paquets qui sortent de votre serveur.



**Étape 1 – Installation du pare-feu Iptables Linux**

1. **Installation d’Iptables**

Iptables est préinstallé dans presque toutes les distributions Linux. Mais si vous ne l’avez pas installé sur votre système [Ubuntu](https://www.hostinger.fr/tutoriels/git-ubuntu)/Debian :

sudo apt-get update

sudo apt-get install iptables

1. **Vérification du statut Iptables actuel**

Avec cette commande, vous pouvez vérifier l’état de votre configuration Iptables actuelle. L’option Here **-L** est utilisée pour énumérer toutes les règles et l’option **-v**  pour une liste plus fastidieuse. Veuillez noter que ces options sont **sensibles à la casse**.

sudo iptables -L -v

Exemple de résultat :

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

C’est la sortie de la commande ci-dessus. Ici, les trois chaînes sont définies par défaut pour la stratégie **ACCEPT**. Il n’y a actuellement aucune règle pour l’une des chaînes.

Pour rendre ce didacticiel Iptables plus pratique, nous modifierons la chaîne **INPUT** pour filtrer le trafic entrant.

**Étape 2 – Définir les règles de la chaîne**

Définir une règle signifie l’ajouter à la liste (chaîne). Voici la commande Iptables formatée avec des options régulières. Nous ne devons pas toutes les spécifier.

sudo iptables -A -i <interface> -p <protocol (tcp/udp) > -s <source> --dport <port no.> -j <target>

Ici **-A** signifie append. La chaîne se réfère à la chaîne que nous voulons ajouter à nos règles. L’**interface** est l’interface réseau sur laquelle vous souhaitez filtrer le trafic. Le **protocole** se réfère au protocole de réseau des paquets que vous souhaitez filtrer. Vous pouvez également spécifier le **numéro du** **port** sur lequel vous souhaitez filtrer le trafic.

Pour plus d’informations sur la commande Iptables et ses options, vous pouvez consulter [la page principale d’Iptables.](https://ipset.netfilter.org/iptables.man.html)

* **Activation du trafic sur localhost**

Nous voulons que toutes les communications entre les applications et les bases de données sur le serveur se poursuivent comme d’habitude.

sudo iptables -A INPUT -i lo -j ACCEPT

Exemple de résultat :

Chain INPUT (policy ACCEPT 7 packets, 488 bytes)

pkts bytes target prot opt in out source destination

0 0 ACCEPT all -- lo any anywhere anywhere

Ici l’option **– A**  est utilisée pour ajouter la règle à la chaîne **INPUT**, accepter toutes les connexions sur l’interface **lo**. Lo signifie une interface de bouclage. Il est utilisé pour toutes les communications sur l’hôte local, comme les communications entre une base de données et une application Web sur la même machine.

* **Activation des connexions sur le port HTTP, SSH et SSL**

Nous voulons que nos connexions HTTP (port 80), https (port 443), ssh (port 22) habituelles se poursuivent comme d’habitude. Entrez les commandes suivantes pour les activer. Dans les commandes suivantes, nous avons spécifié le protocole avec l’option **-p** et le port correspondant pour chaque protocole avec l’**option -dport** (destination port).

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

Maintenant, toutes les connexions au protocole TCP avec les ports spécifiés seront acceptées.

1. **Filtrage des paquets basés sur la source**

Si vous souhaitez accepter ou rejeter des paquets en fonction de l’adresse IP source ou de la plage d’adresses IP, vous pouvez le spécifier avec l’  option **-s**. Par exemple, pour accepter les paquets à partir de l’adresse 192.168.1.3 –

sudo iptables -A INPUT -s 192.168.1.3 -j ACCEPT

Vous pouvez rejeter des paquets d’une adresse IP avec une commande similaire avec l’option  **DROP**.

sudo iptables -A INPUT -s 192.168.1.3 -j DROP

Si vous souhaitez rejeter des paquets à partir d’une gamme d’adresses IP, vous devez utiliser le  module **Iprange** avec l’option **-m** et spécifier la plage d’adresses IP avec **-src-range.**

sudo iptables -A INPUT -m iprange --src-range 192.168.1.100-192.168.1.200 -j DROP

1. **Rejeter d’autres trafics**

**Remarque :**Il est important de **DROP** tous les autres trafics après avoir défini les règles, car cela empêche l’accès non autorisé à un serveur à partir d’autres ports ouverts.

sudo iptables -A INPUT -j DROP

Cette commande rejette tout trafic entrant autre que les ports mentionnés dans les commandes ci-dessus. Vous pouvez vérifier votre ensemble de règles maintenant avec :

sudo iptables -L -v

1. **Suppression des règles**

Si vous souhaitez supprimer toutes les règles et recommencer de zéro, vous pouvez utiliser la commande Flush.

sudo iptables -F

Cette commande supprime toutes les règles actuelles. Si vous souhaitez supprimer une règle spécifique, vous pouvez le faire avec une option **-D**. D’abord, énumérez toutes les règles avec les chiffres en entrant la commande suivante :

sudo iptables -L --line-numbers

Ensuite, vous obtiendrez une liste de règles avec des nombres.

Chain INPUT (policy ACCEPT)

num target prot opt source destination

1 ACCEPT all -- 192.168.0.4 anywhere

2 ACCEPT tcp -- anywhere anywhere tcp dpt:https

3 ACCEPT tcp -- anywhere anywhere tcp dpt:http

4 ACCEPT tcp -- anywhere anywhere tcp dpt:ssh

Pour supprimer une règle, spécifiez le numéro dans la liste et la chaîne de la règle. Dans notre cas, la chaîne **INPUT** et le numéro **3**.

sudo iptables -D INPUT 3

**Étape 3 – Changements persistants**

Les règles Iptables que nous avons créées sont enregistrées en mémoire. Cela signifie que nous devons les redéfinir au redémarrage. Pour que ces modifications soient persistantes après le redémarrage, utilisez la commande suivante sur les systèmes Ubuntu/Debian:

sudo /sbin/iptables-save

Cette commande sauve les règles actuelles dans le fichier de configuration du système qui est utilisé pour reconfigurer les tableaux au moment du redémarrage. Vous devez exécuter cette commande chaque fois que vous modifiez les règles. Pour désactiver ce pare-feu, il suffit de supprimer toutes les règles et de rendre les modifications persistantes.

sudo iptables -F

sudo /sbin/iptables-save