

**Pare-feu OPNsense**

Projet gsb.org

DUMAS Lucie

Table des matières

[Pare-feu et OPNsense 3](#_Toc167715872)

[Qu’est-ce qu’un pare-feu ? 3](#_Toc167715873)

[OPNsense 3](#_Toc167715874)

[IPtables 5](#_Toc167715875)

[OPNsense 10](#_Toc167715876)

[Mise en place du pare-feu OPNsense 10](#_Toc167715877)

[Configuration d’OPNsense 11](#_Toc167715878)

[Tests 16](#_Toc167715879)

[HTTP 16](#_Toc167715880)

[HTTPS 17](#_Toc167715881)

[ICMP 18](#_Toc167715882)

[SSH 19](#_Toc167715883)

[DNS 20](#_Toc167715884)

[DHCP 21](#_Toc167715885)

[BDD 22](#_Toc167715886)

[FTPS 23](#_Toc167715887)

[SAMBA 24](#_Toc167715888)

# Pare-feu et OPNsense

## Qu’est-ce qu’un pare-feu ?

Un pare-feu, également connu sous le nom de firewall en anglais, est un dispositif de sécurité informatique conçu pour surveiller, filtrer et contrôler le trafic réseau, en fonction de règles prédéfinies. L'objectif principal d'un pare-feu est de protéger un réseau informatique, comme celui d'une entreprise ou d'un particulier, en empêchant l'accès non autorisé ou en bloquant les communications indésirables.

Il existe deux types principaux de pare-feu :

* Pare-feu matériel : Il s'agit d'un dispositif physique dédié, généralement installé entre le réseau local et l'Internet. Les pare-feu matériels sont autonomes et agissent comme une barrière entre le réseau interne et les menaces externes. Ils peuvent offrir une protection robuste en filtrant le trafic en fonction d'adresses IP, de ports et de protocoles.
* Pare-feu logiciel : Il s'agit d'un programme logiciel installé sur un ordinateur ou un serveur. Les pare-feu logiciels peuvent être configurés pour contrôler le trafic entrant et sortant de cet appareil particulier. Ils sont souvent utilisés pour protéger les ordinateurs individuels et peuvent être intégrés dans des solutions de sécurité plus larges.

Les pare-feu peuvent être configurés pour autoriser ou bloquer différents types de trafic en fonction de règles spécifiques définies par l'administrateur réseau. Ces règles peuvent être basées sur des adresses IP, des ports, des protocoles ou d'autres critères. Les pare-feu sont un élément essentiel de la sécurité informatique, contribuant à protéger les réseaux contre les attaques malveillantes telles que les tentatives d'intrusion, les virus, les vers et autres menaces en ligne.

## OPNsense

OPNsense est une distribution open-source de pare-feu et de routeur basée sur le système d'exploitation FreeBSD. Elle est conçue pour fournir des fonctionnalités avancées de sécurité réseau, de gestion du trafic et d'administration système. OPNsense offre une interface utilisateur basée sur le web pour simplifier la configuration et la gestion des fonctionnalités du pare-feu.

Voici quelques caractéristiques importantes d'OPNsense :

* Pare-feu avancé : OPNsense propose un ensemble complet de fonctionnalités de pare-feu, y compris la prise en charge de la prévention d'intrusion (IPS), de la détection de logiciels malveillants, du filtrage de contenu, de la gestion des règles de pare-feu et plus encore.
* VPN (Virtual Private Network) : Il offre la possibilité de mettre en place des connexions VPN, permettant ainsi de sécuriser les communications réseau à travers Internet.
* Proxy : OPNsense peut être configuré pour agir en tant que serveur proxy, offrant ainsi des fonctionnalités telles que le filtrage web et la mise en cache.

OPNsense fait donc office de pare-feu logiciel.

# IPtables

Afin de configurer notre pare-feu, nous créons un script IPtables. Ce script servira de base à la création de notre pare-feu avec OPNsense :

#!/bin/bash

iptables -F

iptables -X

iptables -P FORWARD DROP

iptables -P INPUT DROP

iptables -P OUTPUT DROP

#---------------------------------------------------------------------------------------

#-- STATEFULL --

#---------------------------------------------------------------------------------------

iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

#---------------------------------------------------------------------------------------

#-- SSH --

#---------------------------------------------------------------------------------------

#reseau beaup vers notre reseau

iptables -A FORWARD -p tcp -s 10.187.20.0/24 -d 10.31.176.0/20 --dport 22 -j ACCEPT

#autoriser les connexion ssh entre lan et dmz

iptables -A FORWARD -p tcp -s 10.31.176.0/20 -d 10.31.176.0/20 --dport 22 -j ACCEPT

#reseau beaup vers routeur

iptables -A INPUT -p tcp -s 10.187.20.0/24 --dport 22 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- ICMP --

#---------------------------------------------------------------------------------------

# autoriser uniquement les echo reply et request

iptables -A FORWARD -p icmp --icmp-type 0 -s 10.31.176.0/20 -j ACCEPT

iptables -A FORWARD -p icmp --icmp-type 8 -s 10.31.176.0/20 -j ACCEPT

#autoriser les pings du reseau de beaup vers notre reseau

iptables -A FORWARD -p icmp --icmp-type 0 -s 10.187.20.0/24 -d 10.31.176.0/20 -j ACCEPT

iptables -A FORWARD -p icmp --icmp-type 8 -s 10.31.176.0/20 -d 10.187.20.0/24 -j ACCEPT

#autoriser les pings de Backuppc vers la DMZ

iptables -A FORWARD -p icmp --icmp-type 0 -s 10.31.177.73/22 -d 10.31.185.0/22 -j ACCEPT

iptables -A FORWARD -p icmp --icmp-type 8 -s 10.31.185.0/22 -d 10.31.177.73/22 -j ACCEPT

iptables -A FORWARD -p icmp --icmp-type 0 -s 10.31.177.73/22 -d 10.31.186.0/22 -j ACCEPT

iptables -A FORWARD -p icmp --icmp-type 8 -s 10.31.186.0/22 -d 10.31.177.73/22 -j ACCEPT

#autoriser les pings vers notre routeur

iptables -A INPUT -p icmp --icmp-type 0 -d 10.187.20.0/24 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type 8 -s 10.31. 179.254/22 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type 8 -s 10.31.187.254/22 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type 8 -s 172.31.176.254/22 -j ACCEPT

#autoriser notre routeur a ping le réseau 10.31.80.0/20

iptables -A INPUT -p icmp --icmp-type 0 -s 10.31.179.254/22 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type 0 -s 10.31.187.254/22 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type 0 -s 172.31.176.254/22 -j ACCEPT

iptables -A OUTPUT -p icmp --icmp-type 8 -d 10.31.176.0/20 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- DNS --

#---------------------------------------------------------------------------------------

#autoriser contact reseau beaup vers dns 1 et 2

iptables -A FORWARD -p udp --dport 53 -s 10.187.20.0/24 -d 10.31.185.53 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.187.20.0/24 -d 10.31.185.54 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.187.20.0/24 -d 10.31.186.53 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.187.20.0/24 -d 10.31.186.54 -j ACCEPT

# DNS Marius

iptables -A FORWARD -p udp --dport 53 -s 10.31.185.53 -d 8.8.8.8 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.185.53 -d 8.8.4.4 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.185.54 -d 8.8.8.8 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.185.54 -d 8.8.4.4 -j ACCEPT

# DNS Lucie

iptables -A FORWARD -p udp --dport 53 -s 10.31.186.53 -d 8.8.8.8 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.186.53 -d 8.8.4.4 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.186.54 -d 8.8.8.8 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.186.54 -d 8.8.4.4 -j ACCEPT

#autoriser le routeur a faire des request vers les dns

iptables -A OUTPUT -p udp --dport 53 -d 10.31.185.53 -j ACCEPT

iptables -A OUTPUT -p udp --dport 53 -d 10.31.185.54 -j ACCEPT

iptables -A OUTPUT -p udp --dport 53 -d 10.31.186.53 -j ACCEPT

iptables -A OUTPUT -p udp --dport 53 -d 10.31.186.54 -j ACCEPT

#autoriser LAN a faire de request DNS

iptables -A FORWARD -p udp --dport 53 -s 10.31.176.0/22 -d 10.31.185.53 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.176.0/22 -d 10.31.185.54 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.176.0/22 -d 10.31.186.53 -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -s 10.31.176.0/22 -d 10.31.186.54 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- HTTP --

#---------------------------------------------------------------------------------------

#autoriser les request http vers nos serveur web

iptables -A FORWARD -p tcp --dport 80 -s 10.187.20.0/24 -d 10.31.185.80 -j ACCEPT

iptables -A FORWARD -p tcp --dport 80 -s 10.187.20.0/24 -d 10.31.177.80 -j ACCEPT

iptables -A FORWARD -p tcp --dport 80 -s 10.187.20.0/24 -d 10.31.186.80 -j ACCEPT

iptables -A FORWARD -p tcp --dport 80 -s 10.187.20.0/24 -d 10.31.178.80 -j ACCEPT

# backuppc

iptables -A FORWARD -p tcp --dport 80 -s 10.187.20.0/24 -d 10.31.177.73 -j ACCEPT

iptables -A FORWARD -p tcp --dport 80 -s 10.187.20.0/24 -d 10.31.178.73 -j ACCEPT

#autoriser nos serveur web a avoir internet

iptables -A FORWARD -p tcp --dport 80 -s 10.31.184.0/22 ! -d 10.31.176.0/22 -j ACCEPT

iptables -A FORWARD -p tcp --dport 80 -s 10.31.176.0/22 ! -d 10.31.184.0/22 -j ACCEPT

#accès internet sur notre réseau

iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- HTTPS --

#---------------------------------------------------------------------------------------

#autoriser les request http vers nos serveur web

iptables -A FORWARD -p tcp --dport 443 -s 10.187.20.0/24 -d 10.31.185.80 -j ACCEPT

iptables -A FORWARD -p tcp --dport 443 -s 10.187.20.0/24 -d 10.31.177.80 -j ACCEPT

iptables -A FORWARD -p tcp --dport 443 -s 10.187.20.0/24 -d 10.31.186.80 -j ACCEPT

iptables -A FORWARD -p tcp --dport 443 -s 10.187.20.0/24 -d 10.31.178.80 -j ACCEPT

#autoriser nos serveur web a avoir internet

iptables -A FORWARD -p tcp --dport 443 -s 10.31.184.0/22 ! -d 10.31.176.0/22 -j ACCEPT

iptables -A FORWARD -p tcp --dport 443 -s 10.31.176.0/22 ! -d 10.31.184.0/22 -j ACCEPT

# backuppc

iptables -A FORWARD -p tcp --dport 443 -s 10.187.20.0/24 -d 10.31.177.73 -j ACCEPT

iptables -A FORWARD -p tcp --dport 443 -s 10.187.20.0/24 -d 10.31.178.73 -j ACCEPT

#accès internet sur notre routeur

iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- PROXMOX --

#---------------------------------------------------------------------------------------

iptables -A FORWARD -p tcp --dport 8006 -s 10.187.20.0/24 -d 10.31.176.1 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- FTP/S --

#---------------------------------------------------------------------------------------

#autoriser les transfert ftp

# Marius

iptables -A FORWARD -p tcp --dport 21 -s 10.187.20.0/24 -d 10.31.185.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 20 -d 10.187.20.0/24 -s 10.31.185.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 21 -s 10.187.20.0/24 -d 10.31.185.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 20 -d 10.187.20.0/24 -s 10.31.185.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 21 -s 10.187.20.0/24 -d 10.31.185.16 -j ACCEPT

iptables -A FORWARD -p tcp --dport 20 -d 10.187.20.0/24 -s 10.31.185.16 -j ACCEPT

# Lucie

iptables -A FORWARD -p tcp --dport 21 -s 10.187.20.0/24 -d 10.31.186.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 20 -d 10.187.20.0/24 -s 10.31.186.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 21 -s 10.187.20.0/24 -d 10.31.186.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 20 -d 10.187.20.0/24 -s 10.31.186.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 21 -s 10.187.20.0/24 -d 10.31.186.16 -j ACCEPT

iptables -A FORWARD -p tcp --dport 20 -d 10.187.20.0/24 -s 10.31.186.16 -j ACCEPT

#autoriser les transfert ftps

# Marius

iptables -A FORWARD -p tcp --dport 990 -s 10.187.20.0/24 -d 10.31.185.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 989 -d 10.187.20.0/24 -s 10.31.185.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 990 -s 10.187.20.0/24 -d 10.31.185.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 989 -d 10.187.20.0/24 -s 10.31.185.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 990 -s 10.187.20.0/24 -d 10.31.185.16 -j ACCEPT

iptables -A FORWARD -p tcp --dport 989 -d 10.187.20.0/24 -s 10.31.185.16 -j ACCEPT

# Lucie

iptables -A FORWARD -p tcp --dport 990 -s 10.187.20.0/24 -d 10.31.186.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 989 -d 10.187.20.0/24 -s 10.31.186.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 990 -s 10.187.20.0/24 -d 10.31.186.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 989 -d 10.187.20.0/24 -s 10.31.186.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 990 -s 10.187.20.0/24 -d 10.31.186.16 -j ACCEPT

iptables -A FORWARD -p tcp --dport 989 -d 10.187.20.0/24 -s 10.31.186.16 -j ACCEPT

#gère le mode passif de ftp/ftps

# Marius

iptables -A FORWARD -p tcp --dport 55000:60000 -s 10.187.20.0/24 -d 10.31.185.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 55000:60000 -s 10.187.20.0/24 -d 10.31.185.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 55000:60000 -s 10.187.20.0/24 -d 10.31.185.16 -j ACCEPT

# Lucie

iptables -A FORWARD -p tcp --dport 55000:60000 -s 10.187.20.0/24 -d 10.31.186.20 -j ACCEPT

iptables -A FORWARD -p tcp --dport 55000:60000 -s 10.187.20.0/24 -d 10.31.186.15 -j ACCEPT

iptables -A FORWARD -p tcp --dport 55000:60000 -s 10.187.20.0/24 -d 10.31.186.16 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- SAMBA --

#---------------------------------------------------------------------------------------

iptables -A FORWARD -p tcp --dport 445 -s 10.187.20.0/24 -d 10.31.177.13 -j ACCEPT

iptables -A FORWARD -p tcp --dport 445 -s 10.187.20.0/24 -d 10.31.178.13 -j ACCEPT

#---------------------------------------------------------------------------------------

#-- DB --

#---------------------------------------------------------------------------------------

iptables -A FORWARD -p tcp --dport 3306 -s 10.31.185.80 -d 10.31.177.33 -j ACCEPT

iptables -A FORWARD -p tcp --dport 3306 -s 10.31.186.80 -d 10.31.178.33 -j ACCEPT

Nous pouvons exécuter le script et vérifier que toutes les communications puissent aboutir une à une.

# OPNsense

## Mise en place du pare-feu OPNsense

Dans un premier temps, nous commençons par créer une clé bootable à l'aide du logiciel Rufus. Nous téléchargeons l'image sur le site suivant : https://opnsense.org/download/.

Avant de reconfigurer notre routeur, nous notons les correspondances entre nos interfaces réseaux, nos adresses MAC et nos adresses IP afin de pouvoir les reconfigurer plus tard lors de l'installation du pare feu :

* enp2s0 (WAN) : 172.31.176.254/16 | fc:aa:14:52:b3:24
* enp4s0 (LAN) : 10.31.179.254/22 | 64:ee:b7:23:b1:87
* enp5s0 (DMZ) : 10.31.187.254/22 | 64:ee:b7:23:b1:59

Nous pouvons commencer à faire l'installation d'OPNsense sur notre routeur à l'aide de notre clé bootable. Nous lançons OPNsense. Par défaut, le compte utilisateur est root/opnsense. Nous nous connectons sur le terminal et choisissons l'option “Shell” (8). Nous pouvons installer OPNsense en entrant la commande ci-dessous :

apt update && apt upgrade

opnsense-installer

Plusieurs étapes sont à compléter :

1. Sélection du clavier :

2. Choix du système de fichiers (ZFS)

3. Partitionnement

4. Sélection du disque

5. Suppression des anciennes données du disque

6. Autoriser/Refuser le swap

7. Changement du mot de passe

8. Finalisation de l'installation

Nous pouvons maintenant configurer nos interfaces en sélectionnant l'option “Assign interfaces” (1). Nous commençons par assigner les interfaces à leurs nom en faisant la correspondance avec leur nom et leur adresse MAC. Ainsi :

re0 = WAN

re1 = LAN

opt1 = DMZ

Il est possible durant l'assignation des interfaces d'en rajouter une nouvelle étant donné que seulement deux interfaces (re0 et re1) sont disponibles au début de la configuration.

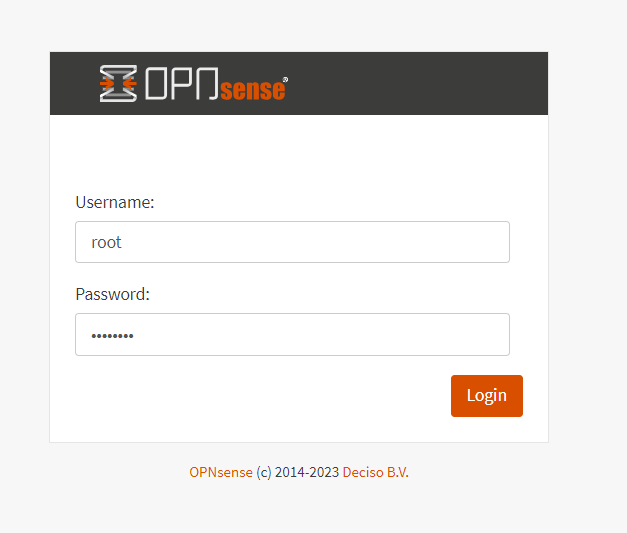
Nous devons maintenant assigner nos adresses IP à nos interfaces. Pour ce faire, nous choisissons l'option “Set interface(s) IP address” (2). Nous assignons à chaque interface son adresse IP ainsi que son masque, et ajoutons à l'interface WAN sa passerelle par défaut (172.31.0.1/16).

Par défaut, le pare-feu est activé et bloque tous les flux. Pour accéder à l'interface web, nous devons momentanément désactiver le pare-feu :

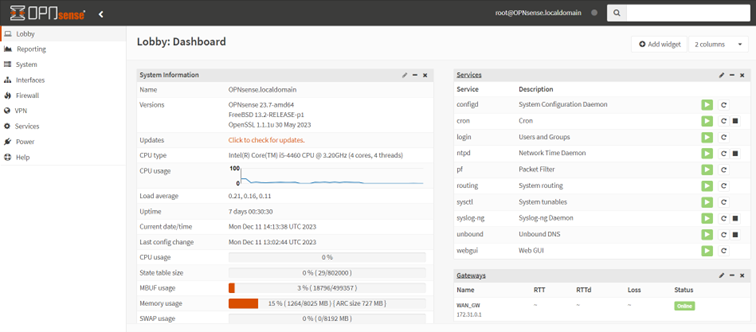
pfctl -d

La commande pour réactiver le pare-feu est « pfctl -e ».

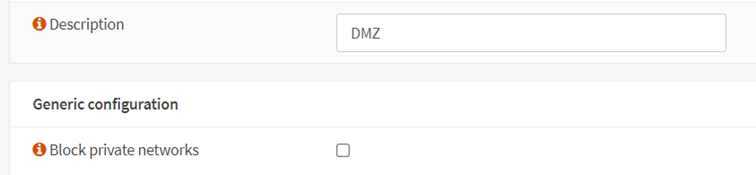
## Configuration d’OPNsense

 Une fois l’installation terminée, nous pouvons accéder à l’interface web d’OPNsense à l’adresse suivante : https://10.31.179.254

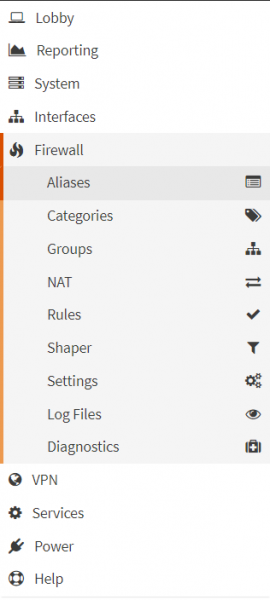
Les identifiants sont ceux que nous avons changé lors de l’installation du pare-feu, donc root/opnsense. Voici l’interface de gestion du pare-feu :



Dans un premier temps, nous devons changer le nom de l'interface reliée au réseau DMZ. Pour cela, nous sélectionnons “Interfaces”, puis l'option “[NOM DE L'INTERFACE]”. Nous en changeons la descriptions et nous décochons la case “Block private networks”. Cela nous permettra de pouvoir autoriser les communications entre nos réseaux ayant des adresses privées. Nous devons décocher cette case pour toutes les interfaces.



Nous créons ensuite de nouveaux alias qui permettront de sélectionner plusieurs machines pour une seule et même règle. Pour ce faire, nous sélectionnons “Firewall”, puis l'option “Aliases” :



Nous créerons de nouveaux aliases qui regrouperont certaines machines afin d'optimiser nos règles :

- Beaupeyrat : 10.187.20.0/24

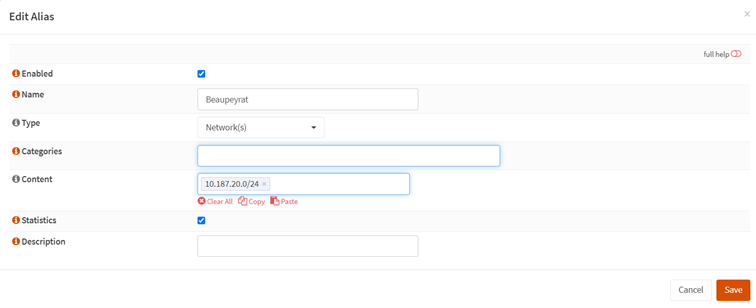
- FTP : 10.31.185.20, 10.31.185.15, 10.31.185.16, 10.31.186.20, 10.31.186.15, 10.31.186.16

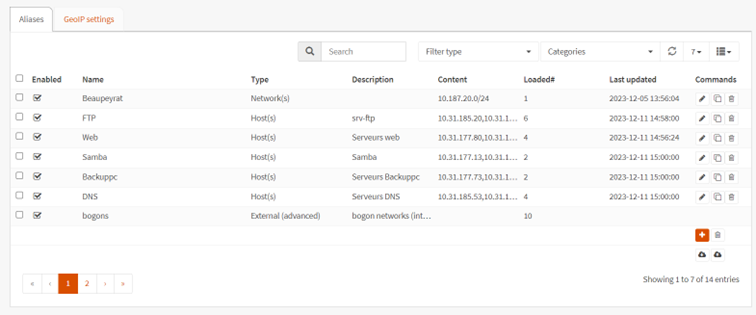
- Web : 10.31.177.80, 10.31.178.80, 10.31.185.80, 10.31.186.80

- Samba : 10.31.177.13, 10.31.178.13

- Backuppc : 10.31.177.73, 10.31.178.73

- DNS : 10.31.185.53, 10.31.185.54, 10.31.186.53, 10.31.186.54



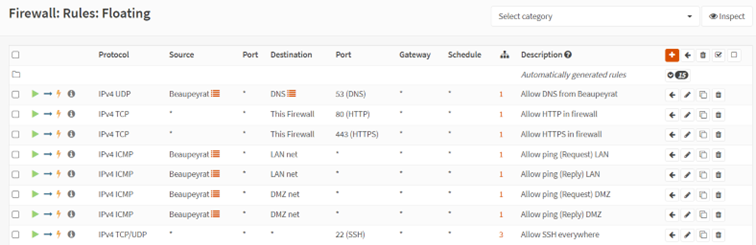
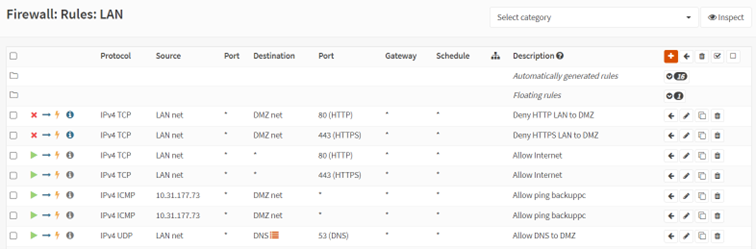


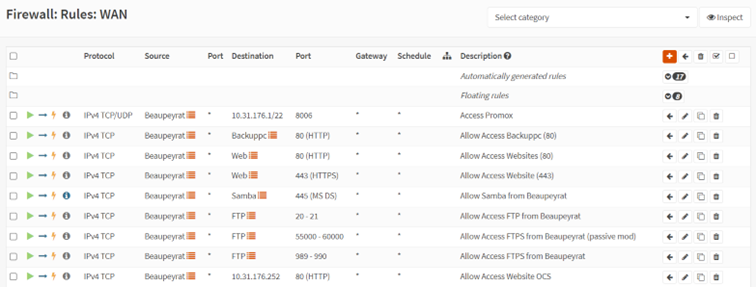
Nous pouvons maintenant écrire nos règles de pare-feu. Toujours dans l'onglet “Firewall”, nous sélectionnons l'option “Rules”. L'option “Floating” correspond aux règles pouvant être affectées à plusieurs interfaces à la fois. Il existe une option pour chaque interface. Nous sélectionnons l'option “Floating”. Nous pouvons créer une nouvelle règle. La règle que nous allons créer sera celle autorisant le réseau de Beaupeyrat (interface WAN) à communiquer avec les DNS (interface DMZ). Nous devons ainsi :

* Activer l'option “Qwick” permettant d'exécuter une action lorsque cette règle est utilisée
* Sélectionner l'Interface concernée (WAN)
* Sélectionner la direction (IN)
* Sélectionner le protocol (UDP)
* Sélectionner une source (l'alias “Beaupeyrat”)
* Sélectionner une destination (l'alias “DNS”)
* Renseigner une catégorie (Allow DNS)
* Renseigner une description (Allow DNS from Beaupeyrat)

 Un message apparait alors nous informant de la modification de règles. Nous cliquons sur le bouton “Apply changes” :

Nous pouvons ainsi créer toutes nos règles pour chaque interface :

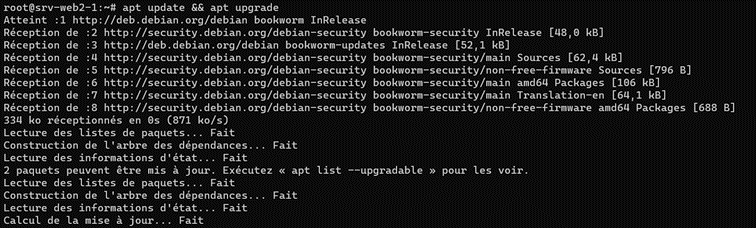


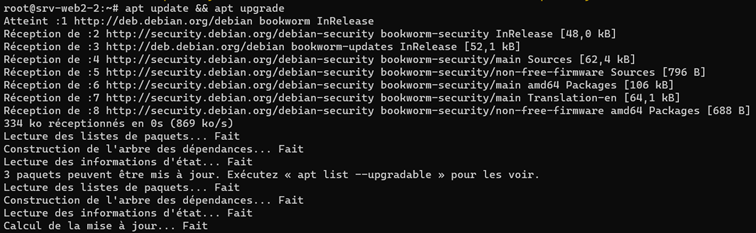


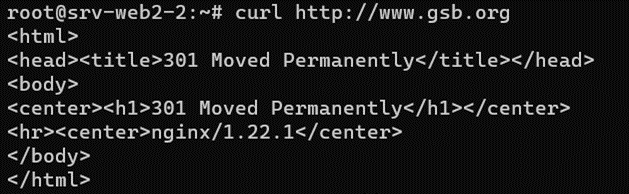
# Tests

## HTTP

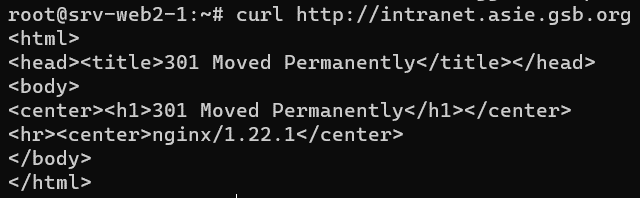
La DMZ doit avoir accès à Internet :



 La LAN doit avoir accès à Internet :

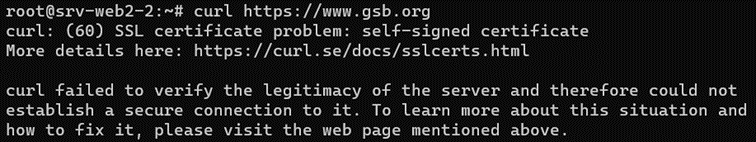
 La LAN ne doit pas avoir accès aux machines de la DMZ en utilisant le protocole HTTP. Voici à quoi ressemble une requête CURL de la LAN vers la DMZ lorsque les flux ne sont pas filtrés :

 Voici les résultats de la requête CURL de la LAN vers la DMZ une fois que les flux sont filtrés :

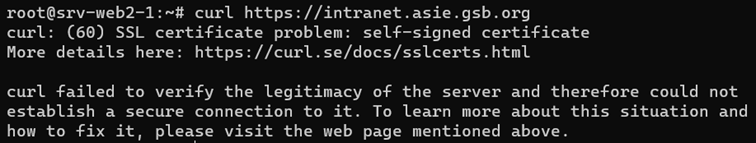
 La DMZ ne doit pas avoir accès aux machines de la LAN en utilisant le protocole HTTP. Voici à quoi ressemble une requête CURL de la DMZ vers la LAN lorsque les flux ne sont pas filtrés :

 Voici les résultats de la requête CURL de la DMZ vers la LAN une fois que les flux sont filtrés :

## HTTPS

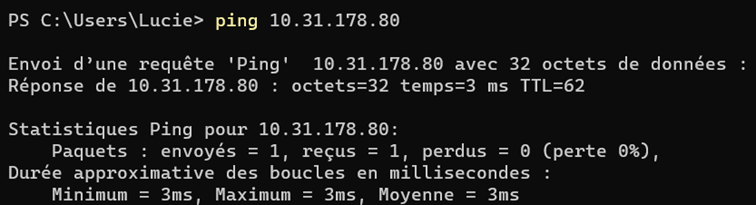
 La LAN ne doit pas avoir accès aux machines de la DMZ en utilisant le protocole HTTPS. Voici à quoi ressemble une requête CURL de la LAN vers la DMZ lorsque les flux ne sont pas filtrés :

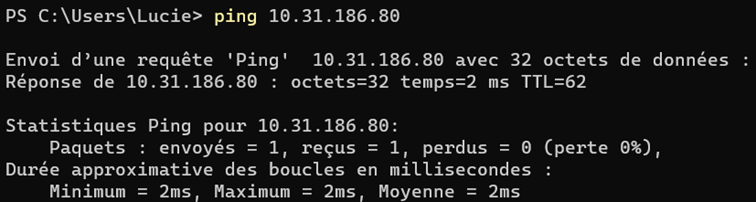
 Voici les résultats de la requête CURL de la LAN vers la DMZ une fois que les flux sont filtrés :

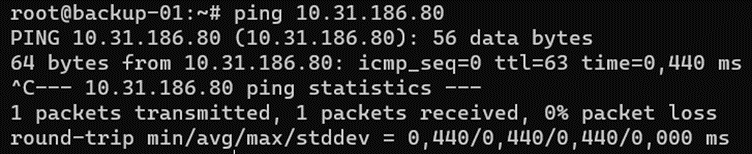
 La DMZ ne doit pas avoir accès aux machines de la LAN en utilisant le protocole HTTPS. Voici à quoi ressemble une requête CURL de la DMZ vers la LAN lorsque les flux ne sont pas filtrés :

 Voici les résultats de la requête CURL de la DMZ vers la LAN une fois que les flux sont filtrés :

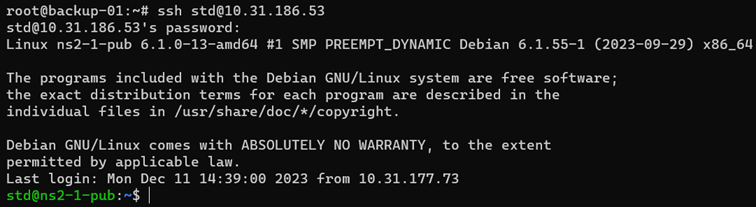
## ICMP

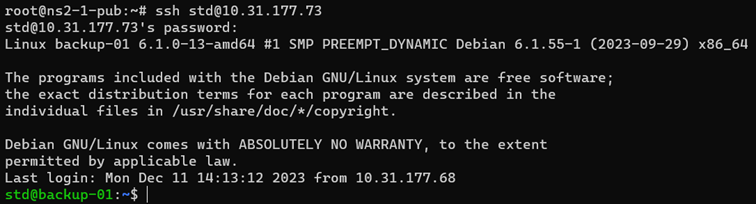
 Les requêtes ICMP request et reply doivent permettre depuis le réseau Beaupeyrat de ping la LAN :

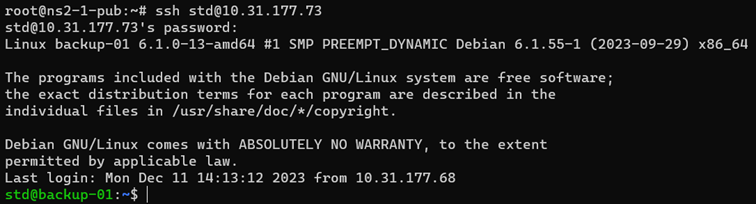
 Les requêtes ICMP request et reply doivent permettre depuis le réseau Beaupeyrat de ping la DMZ :

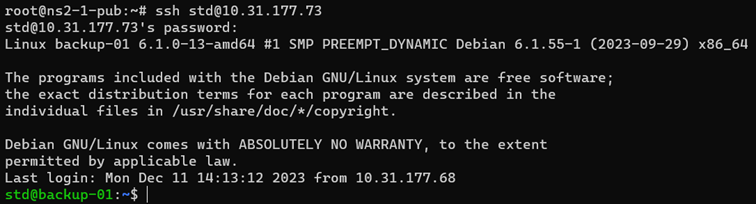
 Les requêtes ICMP request et reply doivent permettre à la machine Backuppc de contacter la DMZ :

## SSH

 Le réseau LAN doit avoir accès au réseau DMZ :

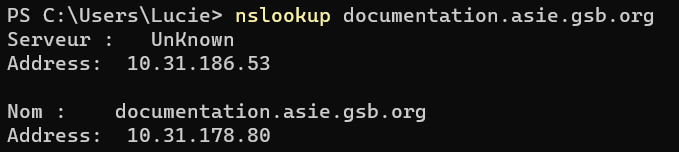
 Le réseau DMZ doit avoir accès au réseau LAN :

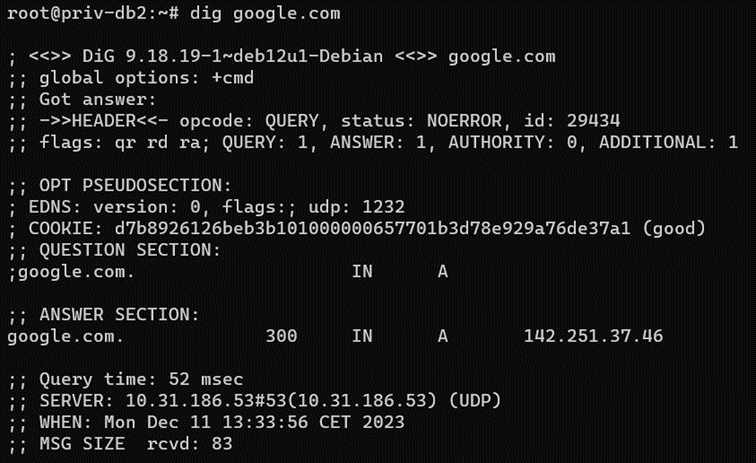
 Le réseau de Beaupeyrat doit avoir accès au réseau DMZ :

 Le réseau de Beaupeyrat doit avoir accès au réseau LAN :

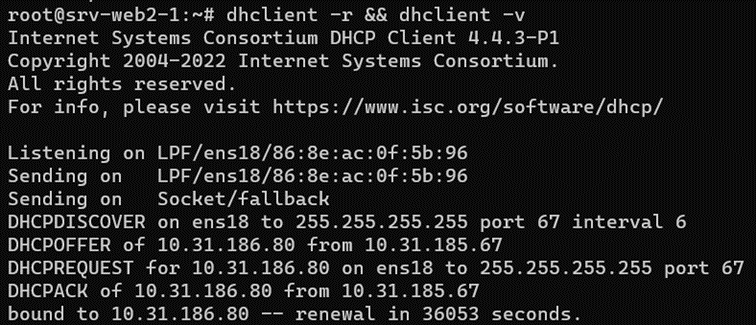
## DNS

Les DNS doivent pouvoir être joignables depuis le réseau de Beaupeyrat :

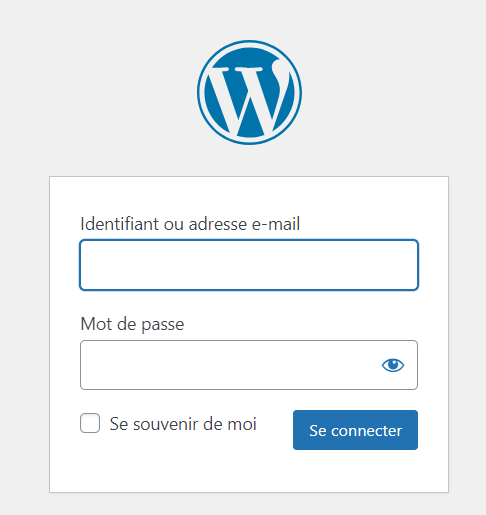


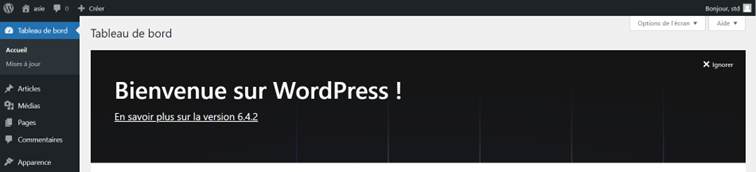
 Les DNS doivent pouvoir être joignables depuis la LAN :

## DHCP

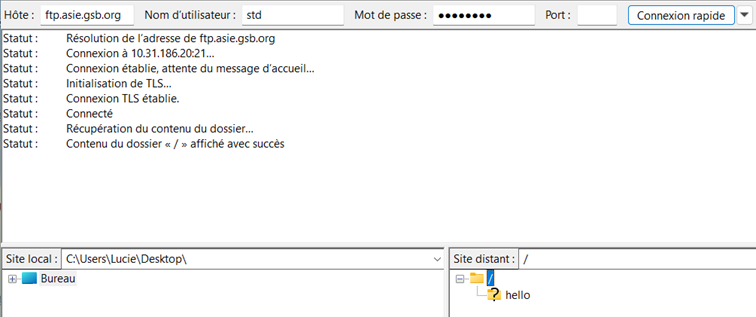
 La DMZ a besoin de contacter le DHCP situé sur la LAN :

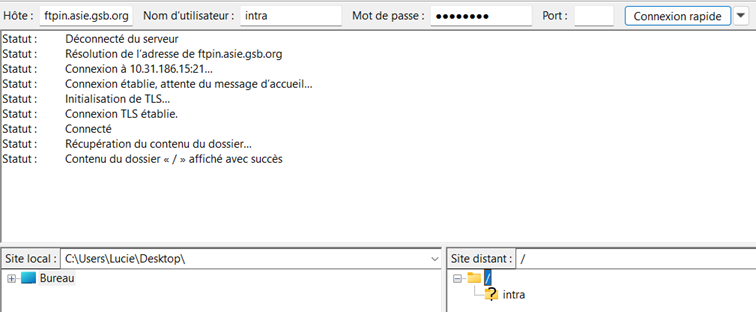
## BDD

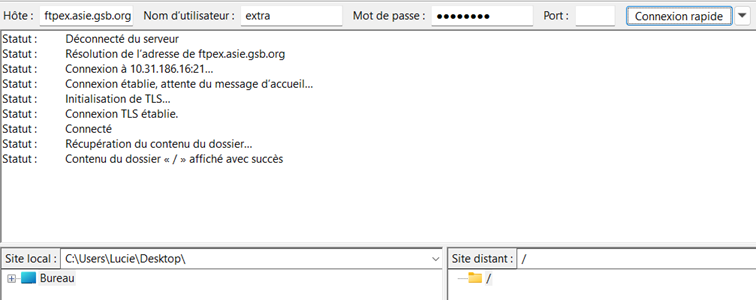
 Le serveur web de la DMZ doit pouvoir contacter la base de données. Pour vérifier la règle, nous pouvons nous rendre dans l'interface d'administration de WordPress via le lien suivant : https://www.asie.gsb.org/wp-admin. Les identifiants du site sont std et password (pour les sites du serveur 10.31.186.80) :



## FTPS

 Le réseau de Beaupeyrat doit avoir accès au serveur de fichiers FTP :





## SAMBA

Le serveur de fichiers Samba doit être accessible depuis le réseau de Beaupeyrat :

