



COMPTE RENDU DE PROJET

Projet : Sécurité Réseaux – Sécurisation d'une infrastructure réseau

Rapport rédigé & produit par : **Kévin MOREAU**

Professeur responsable : **Julien BREYAUT**

École Nationale Supérieure d'Ingénieurs de Bretagne Sud

Apprenti - Ingénieur Expert Cyberdéfense

2^{ème} année

Promotion

AIRBUS

Table des matières

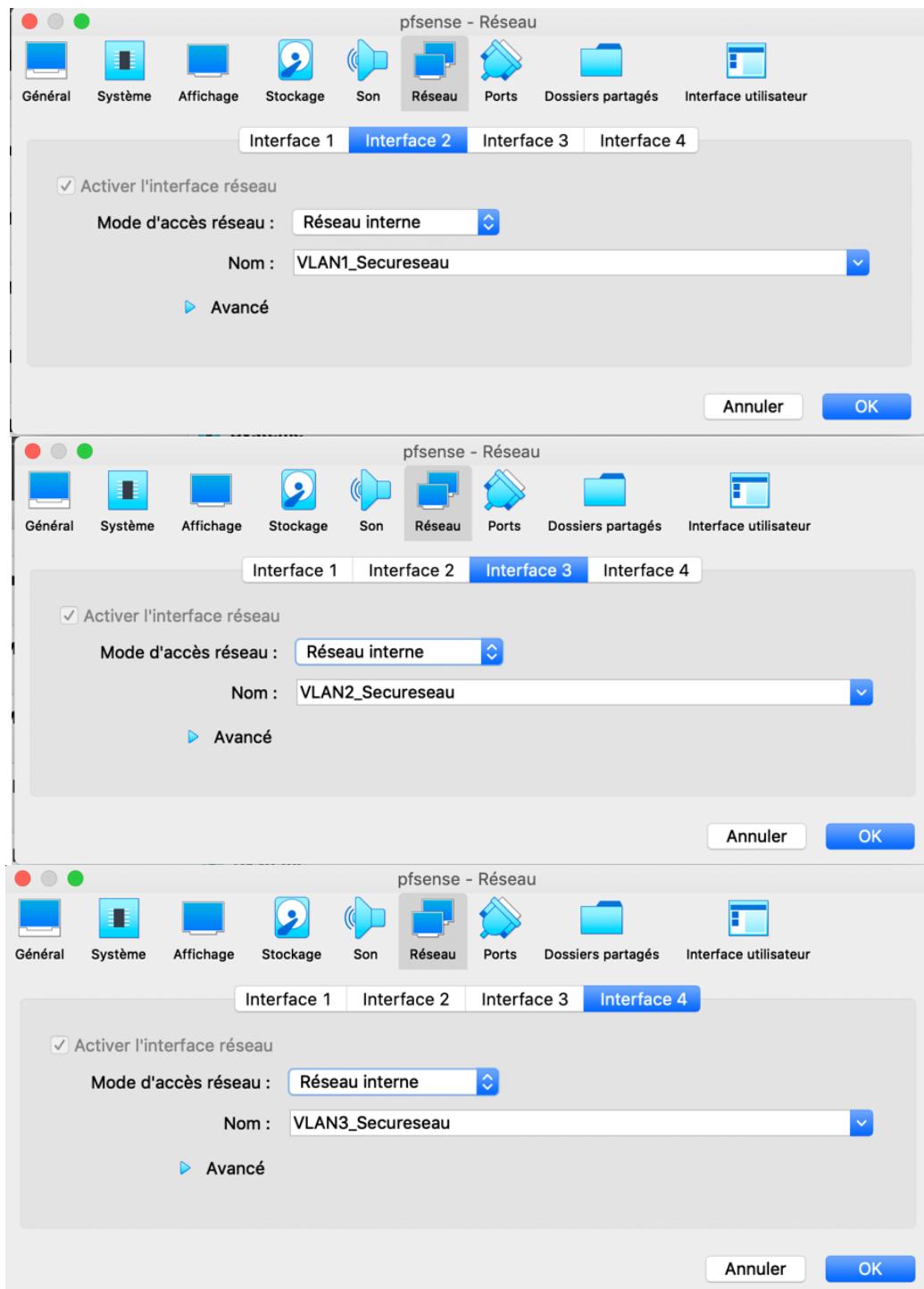
COMPTE RENDU DE PROJET	1
1. Définition des différents VLAN à partir du pare-feu.....	3
2. Configuration de l'interface WAN	5
3. Configuration des deux autres interfaces du pfSense	7
4. Configuration raffinée de sécurité du réseau.....	11
4.1. Flux sortants	12
4.1.1. LAN vers Internet en DNS.....	12
4.1.2. DMZ RELAY vers Internet en HTTP, HTTPS, DNS	13
4.1.3. DMZ RELAY vers Internet en HTTP, HTTPS, DNS	13
Après constat, il semble que la destination	14
4.1.4. DMZ WEB vers Internet en DNS.....	15
4.2. Flux entrants.....	16
4.2.1. HTTP, HTTPS, SMTP vers SRV_RELAI.....	16
4.3. Flux internes	17
4.3.1. SSH depuis PC client vers SRV WEB et SRV RELAY	17
4.3.2. HTTP, HTTPS depuis le PC client vers le SRV_WEB.....	19
4.3.3. HTTP, HTTPS depuis le SRV_Relais vers le SRV_WEB	20
4.3.4. PROXY depuis le PC Client vers SRV_Relais.....	21
4.3.5. PROXY depuis le SRV_Web vers SRV_Relais	23
4.4. Règles de NAT	24
4.4.1. Flux sortants : NAT dynamique en utilisant l'@IP de l'interface WAN pour le trafic sortant depuis le LAN, la DMZ RELAY et la DMZ WEB	25
4.4.2. Flux entrants : NAT statique pour le HTTP, HTTPS, SMTP vers SRV RELAY	26
4.5. Matrice des flux autorisés finale	28
4.6. Configuration du proxy	30
4.6.1. Configuration Squid sur SRV RELAY.....	30
4.6.2. Configuration Squidguard sur SRV RELAY	32
4.7. Configuration du reverse proxy	35
4.7.1. Configuration HAProxy sur SRV RELAY.....	35
4.8. UTM PfSense.....	39
4.8.1. Installer les paquages nécessaires afin d'implémenter les services de proxy et reverse proxy sur le firewall PfSense	39
4.8.2. Configurer des services sur le firewall.....	41
4.8.3. Installer les paquages nécessaires afin d'implémenter un service IPS/IDS	43
4.8.4. Configurer l'IPS/IDS.....	43
5. Retour d'expérience.....	44
6. Annexes et bibliographies.....	45

1. Définition des différents VLAN à partir du pare-feu

Pour commencer, les étapes préliminaires doivent se faire sous [VirtualBox](#).

On commence par assigner les différents [VLAN](#) aux différentes interfaces de notre pare-feu.

Sous [Virtualbox](#), se rendre dans la configuration de la machine virtuelle puis accéder à l'onglet [Réseau](#). A partir de là, on crée les différents [VLAN](#) pour chaque interface :



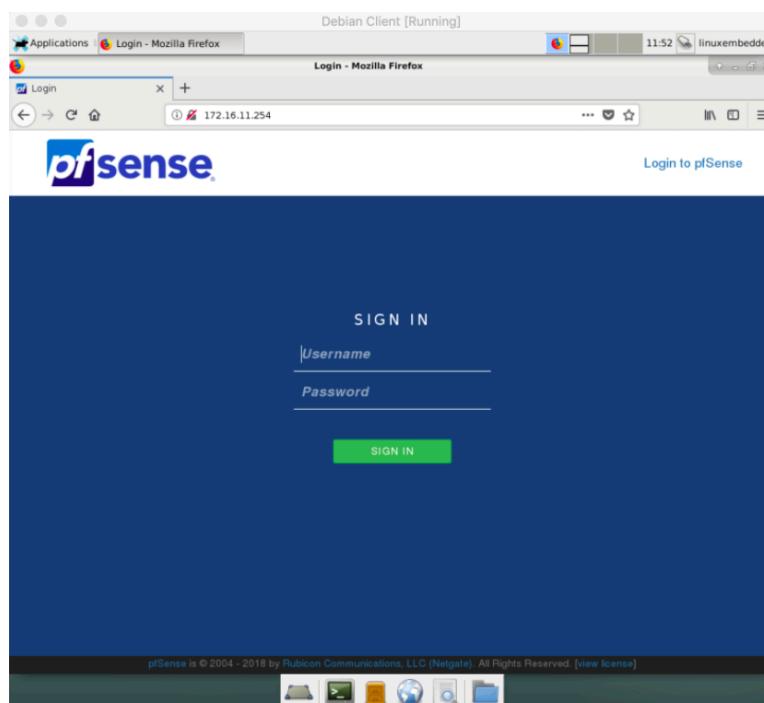
Une fois les adresses IP configurées sur les deux machines, on teste la communication depuis notre machine cliente d'administration :

```
debianClient@linuxembedded:~$ ping 172.16.11.254
```

Concernant la machine cliente, elle dispose de la configuration suivante :

```
debianClient@linuxembedded:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:10:55:16 brd ff:ff:ff:ff:ff:ff
    inet 172.16.11.20/24 brd 172.16.11.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe10:5516/64 scope link
        valid_lft forever preferred_lft forever
```

Enfin, on accède à la page web de configuration du pfSense via l'adresse <http://172.16.11.254>

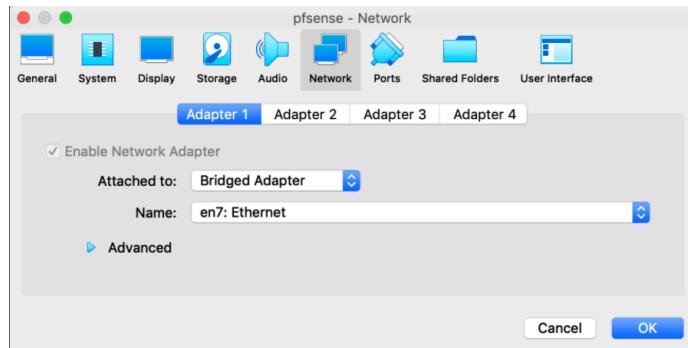


2. Configuration de l'interface WAN

Une fois connecté, on accède au [GUI](#) du pare-feu.

On commence par configurer directement l'interface [WAN](#) pour récupérer une configuration [DHCP](#) depuis la salle.

On configure l'interface 1 depuis [Virtualbox](#) pour récupérer la connexion Ethernet depuis la machine virtuelle :



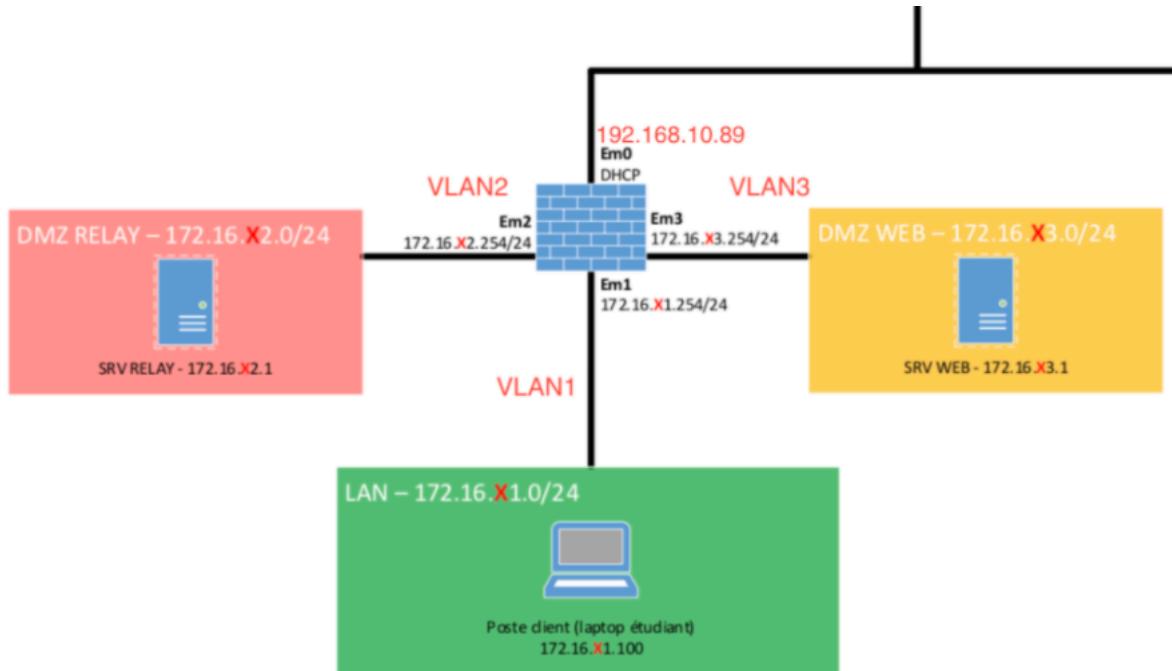
Ensuite, on utilise l'interface graphique pour commencer la configuration de l'interface [WAN](#) :

A screenshot of the pfSense WAN interface configuration page in Mozilla Firefox. The URL is 172.16.11.254/interfaces.php?f=wan. The page has a header with tabs for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The main content area is titled "Interfaces / WAN". It shows the "General Configuration" section with fields for Enable (checked), Description (WAN), IPv4 Configuration Type (DHCP), IPv6 Configuration Type (DHCP6), MAC Address (xx:xx:xx:xx:xx:xx), MTU (1500), MSS (40), and Speed and Duplex (Default). Below this is the "DHCP Client Configuration" section with options for Advanced Configuration and Configuration Override, and fields for Hostname and Alias IPv4 address.

Une fois l'interface configurée, cette interface récupère un bail **DHCP**. On vérifie l'adresse IP et on la garde dans un coin :

```
inet 192.168.10.89 netmask 0xffffffff broadcast 192.168.10.255
```

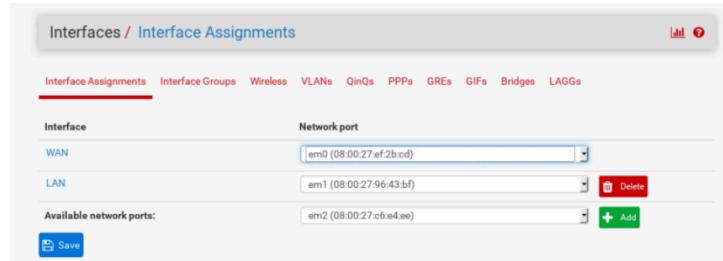
Notre schéma réseau ressemble donc à cela :



3. Configuration des deux autres interfaces du pfSense

Pour finaliser notre réseau, on vient configurer les deux autres [VLAN](#) (2 et 3) pour rajouter les deux DMZ.

Depuis l'interface graphique, on rajoute 2 nouvelles interfaces :



On peut créer nos deux interfaces directement et on les configure avec les adresses attribuées aux interfaces soit :

- Interface [em2](#), renommée [VLANDMZrelais](#) : [172.16.12.254](#)

A screenshot of the pfSense interface configuration for the 'VLANDMZrelais' interface. The 'Description' field is set to 'VLANDMZrelais'. The 'IPv4 Configuration Type' is set to 'Static IPv4'. The 'IPv6 Configuration Type' is set to 'None'. The 'MAC Address' field contains '000000000000'. The 'MTU' field is empty. The 'MSS' field is empty. The 'Speed and Duplex' dropdown is set to 'Default (no preference, typically autoselect)'. The 'Static IPv4 Configuration' section shows the 'IPv4 Address' as '172.16.12.254'. There are also sections for 'Static IPv6 Configuration' and 'Advanced Options' which are currently empty.

- Interface [em3](#), renommée [VLANDMZweb](#) : [172.16.13.254](#)

A screenshot of the pfSense interface configuration for the 'VLANDMZweb' interface. The 'Description' field is set to 'VLANDMZweb'. The 'IPv4 Configuration Type' is set to 'Static IPv4'. The 'IPv6 Configuration Type' is set to 'None'. The 'MAC Address' field contains '000000000000'. The 'MTU' field is empty. The 'MSS' field is empty. The 'Speed and Duplex' dropdown is set to 'Default (no preference, typically autoselect)'. The 'Static IPv4 Configuration' section shows the 'IPv4 Address' as '172.16.13.254'. There are also sections for 'Static IPv6 Configuration' and 'Advanced Options' which are currently empty.

Enfin, on configure nos interfaces sur les postes clients (web et relais) :

The image shows two terminal windows side-by-side. Both windows are titled "GNU nano 2.7.4" and show the same file content: /etc/network/interfaces.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 172.16.13.1/24
    gateway 172.16.13.254
```

The top window is for "web01 [Running]" and the bottom window is for "relay01 [Running]". Both windows show a status bar at the bottom with various keyboard shortcuts. In the relay01 window, the status bar indicates "[15 lignes écrites]".

Une fois les interfaces paramétrées, on ajoute une règle firewall depuis le pfSense pour autoriser les flux ICMP dans les deux sens (entrants / sortants). Cette règle n'est pas directement inscrite dans la matrice des flux demandée. Cependant elle permet de dépanner certains problèmes réseaux de manière simple. Elle sera désactivée par la suite :

Depuis la DMZ Relais, par exemple, cela donne :

Firewall / Rules / VLANDMZRELAIS

No rules are currently defined for this interface.
All incoming connections on this interface will be blocked until pass rules are added. Click the button to add a new rule.

Add Edit Delete Save Separator

Edit Firewall Rule

Action: Pass

Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled: Disable this rule

Interface: VLANDMZRELAIS

Address Family: IPv4

Protocol: ICMP

ICMP Subtypes: any, Alternate Host, Datagram conversion error, Echo reply

Source: Source: any, Invert match:

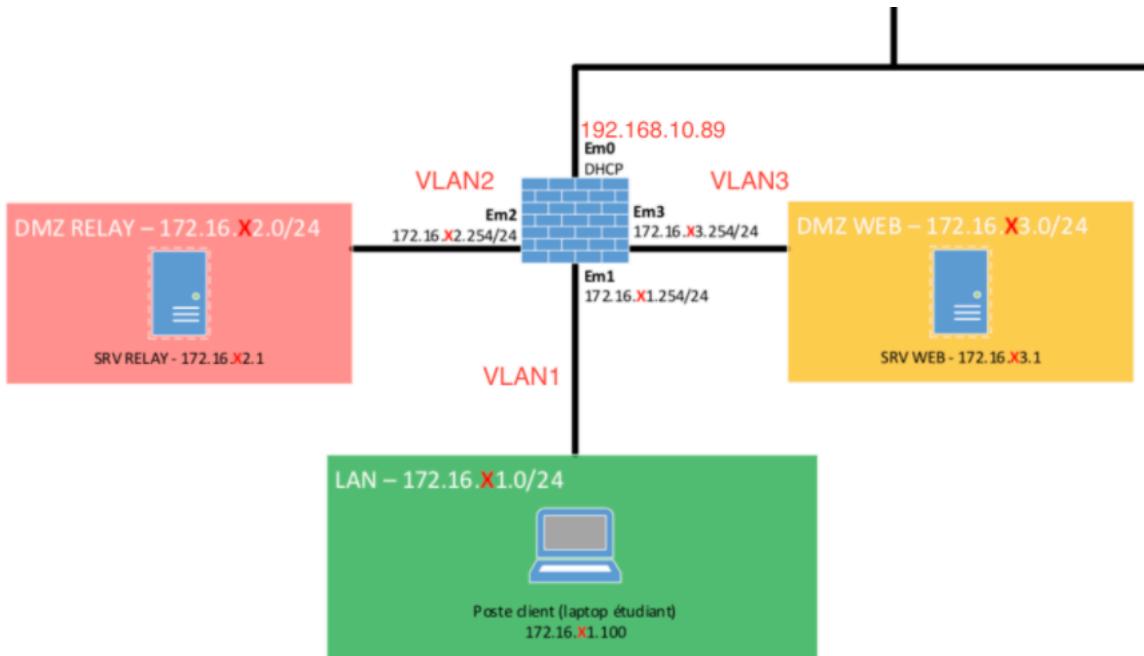
Destination: Destination: any, Invert match:

On établit une règle autorisant le protocole ICMP de n'importe quelle source vers n'importe quelle destination. Ainsi, un client effectuant une requête ping depuis ce VLAN à travers le pare-feu pourra avoir une réponse.

Exemple d'un ping ICMP entre le `client` et la `DMZ relais` :

```
debianClient@linuxembedded:~$ ping 172.16.13.1
PING 172.16.13.1 (172.16.13.1) 56(84) bytes of data.
64 bytes from 172.16.13.1: icmp_seq=1 ttl=63 time=0.766 ms
64 bytes from 172.16.13.1: icmp_seq=2 ttl=63 time=1.13 ms
64 bytes from 172.16.13.1: icmp_seq=3 ttl=63 time=0.739 ms
64 bytes from 172.16.13.1: icmp_seq=4 ttl=63 time=0.730 ms
^C
--- 172.16.13.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3035ms
rtt min/avg/max/mdev = 0.730/0.843/1.137/0.170 ms
```

A partir de là, nous avions bien une connexion réseau sur chaque interface, respectant le schéma réseau initialement donné :



4. Configuration raffinée de sécurité du réseau

Cette partie consiste à répondre aux exigences énoncées dans le TP à savoir :

Filtrage des flux

- **Flux sortants**

- LAN vers Internet en DNS
- DMZ RELAY vers Internet en HTTP, HTTPS, DNS
- DMZ WEB vers Internet en DNS

- **Flux entrants**

- HTTP, HTTPS, SMTP vers SRV RELAY

- **Flux internes**

- SSH depuis PC client vers SRV WEB et SRV RELAY
- HTTP, HTTPS depuis PC client vers SRV WEB
- HTTP, HTTPS depuis SRV RELAY vers SRV WEB
- PROXY depuis PC client vers SRV RELAY
- PROXY depuis SRV WEB vers SRV RELAY

Ci-dessous les captures et tests effectués pour répondre à ces exigences.

4.1. Flux sortants

Ci-dessous l'ensemble des règles concernant les flux sortants, c'est à dire les flux à destination du [WAN](#) (Internet).

4.1.1. LAN vers Internet en DNS

The screenshot shows the 'Edit Firewall Rule' configuration page. The 'Action' dropdown is set to 'Pass'. The 'Interface' is set to 'LAN'. The 'Address Family' is 'IPv4'. The 'Protocol' is 'TCP'. In the 'Source' section, the 'Source' dropdown is set to 'Single host or alias' with 'PosteClient' selected. There is a 'Display Advanced' button. In the 'Destination' section, the 'Destination' dropdown is set to 'any'. The 'Destination Port Range' is set from 'DNS (53)' to 'DNS (53)'. The 'From' and 'To' fields both have 'Custom' selected.

4.1.2. DMZ RELAY vers Internet en HTTP, HTTPS, DNS

Firewall / Rules / Edit

Edit Firewall Rule

Action: Pass

Disabled: Disable this rule

Interface: LAN

Address Family: IPv4

Protocol: TCP

Source

Source: Invert match. LAN net Source Address /

Source Port Range: From: DNS (53) To: DNS (53)

Destination

Destination: Invert match. WAN net Destination Address /

Destination Port Range: From: DNS (53) To: DNS (53)

4.1.3. DMZ RELAY vers Internet en HTTP, HTTPS, DNS

Firewall / Rules / Edit

Edit Firewall Rule

Action: Pass

Disabled: Disable this rule

Interface: VLANDMZRELAIS

Address Family: IPv4

Protocol: TCP

Source

Source: Invert match. VLANDMZRELAIS net Source Address /

Source Port Range: From: HTTP (80) To: HTTP (80)

Destination

Destination: Invert match. WAN net Destination Address /

Destination Port Range: From: HTTP (80) To: HTTP (80)

Firewall / Rules / Edit

Edit Firewall Rule

Action	Pass
Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.	
Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
Interface	VLANDMZRELAIS
Choose the interface from which packets must come to match this rule.	
Address Family	IPv4
Select the Internet Protocol version this rule applies to.	
Protocol	TCP
Choose which IP protocol this rule should match.	
Source	
Source	<input type="checkbox"/> Invert match. <input type="text"/> VLANDMZRELAIS net <input type="button"/> Source Address /
Hide Advanced	
The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.	
Source Port Range	<input type="text"/> HTTPS (443) <input type="button"/> From <input type="text"/> Custom <input type="text"/> To <input type="button"/> Custom
Specify the source port or port range for this rule. The "To" field may be left empty if only filtering a single port.	
Destination	
Destination	<input type="checkbox"/> Invert match. <input type="text"/> WAN net <input type="button"/> Destination Address /
Destination Port Range	<input type="text"/> HTTPS (443) <input type="button"/> From <input type="text"/> Custom <input type="text"/> To <input type="button"/> Custom
Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.	

Firewall / Rules / Edit

Edit Firewall Rule

Action	Pass
Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.	
Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
Interface	VLANDMZRELAIS
Choose the interface from which packets must come to match this rule.	
Address Family	IPv4
Select the Internet Protocol version this rule applies to.	
Protocol	TCP
Choose which IP protocol this rule should match.	
Source	
Source	<input type="checkbox"/> Invert match. <input type="text"/> VLANDMZRELAIS net <input type="button"/> Source Address /
Hide Advanced	
The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.	
Source Port Range	<input type="text"/> DNS (53) <input type="button"/> From <input type="text"/> Custom <input type="text"/> To <input type="button"/> Custom
Specify the source port or port range for this rule. The "To" field may be left empty if only filtering a single port.	
Destination	
Destination	<input type="checkbox"/> Invert match. <input type="text"/> WAN net <input type="button"/> Destination Address /
Destination Port Range	<input type="text"/> DNS (53) <input type="button"/> From <input type="text"/> Custom <input type="text"/> To <input type="button"/> Custom
Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.	

Après constat, il semble que la destination **WAN net** ne fonctionne pas. En effet, aucun flux ne transite depuis le relais vers Internet. La solution est de mettre la destination en **any** permettant au relais d'accéder au **WAN**.

4.1.4. DMZ WEB vers Internet en DNS

Edit Firewall Rule

Action	Pass
Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.	
Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
Interface	VLANDMZWEB
Choose the interface from which packets must come to match this rule.	
Address Family	IPv4
Select the Internet Protocol version this rule applies to.	
Protocol	TCP
Choose which IP protocol this rule should match.	
Source	
Source	<input type="checkbox"/> Invert match. VLANDMZWEB net Source Address /
Hide Advanced The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.	
Source Port Range	From: DNS (53) To: DNS (53) Custom Custom
Specify the source port or port range for this rule. The "To" field may be left empty if only filtering a single port.	
Destination	
Destination	<input type="checkbox"/> Invert match. any Destination Address /
Destination Port Range	From: DNS (53) To: DNS (53) Custom Custom
Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.	

4.2. Flux entrants

Ci-dessous l'ensemble des règles concernant les flux entrants, c'est à dire les flux en provenance du **WAN** (Internet).

4.2.1. HTTP, HTTPS, SMTP vers SRV_RELAIIS

Un **alias** est utilisé pour regrouper l'ensemble de ces ports :

Firewall Aliases Ports	
Name	Values
HTTPS_HTTP_SMTP	80, 443, 25

Firewall / Rules / Edit

Edit Firewall Rule

Action: Pass

Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled: Disable this rule
Set this option to disable this rule without removing it from the list.

Associated filter rule: This is associated with a NAT rule.
Editing the interface, protocol, source, or destination of associated filter rules is not permitted.
[View the NAT rule](#)

Interface: WAN
Choose the interface from which packets must come to match this rule.

Address Family: IPv4
Select the Internet Protocol version this rule applies to.

Protocol: TCP
Choose which IP protocol this rule should match.

Source

Source: Invert match. any Source Address /

[Display Advanced](#)
The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.

Destination

Destination: Invert match. Single host or alias ServeurRelais /

Destination Port Range: (other) From Custom (other) To Custom HTTPS_HTTP_SMTP
Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

4.3. Flux internes

Ci-dessous l'ensemble des règles concernant les flux internes, c'est à dire les flux en provenance et à destination des différents [VLANS](#) ([VLAN1](#), [VLAN2](#) & [VLAN3](#))

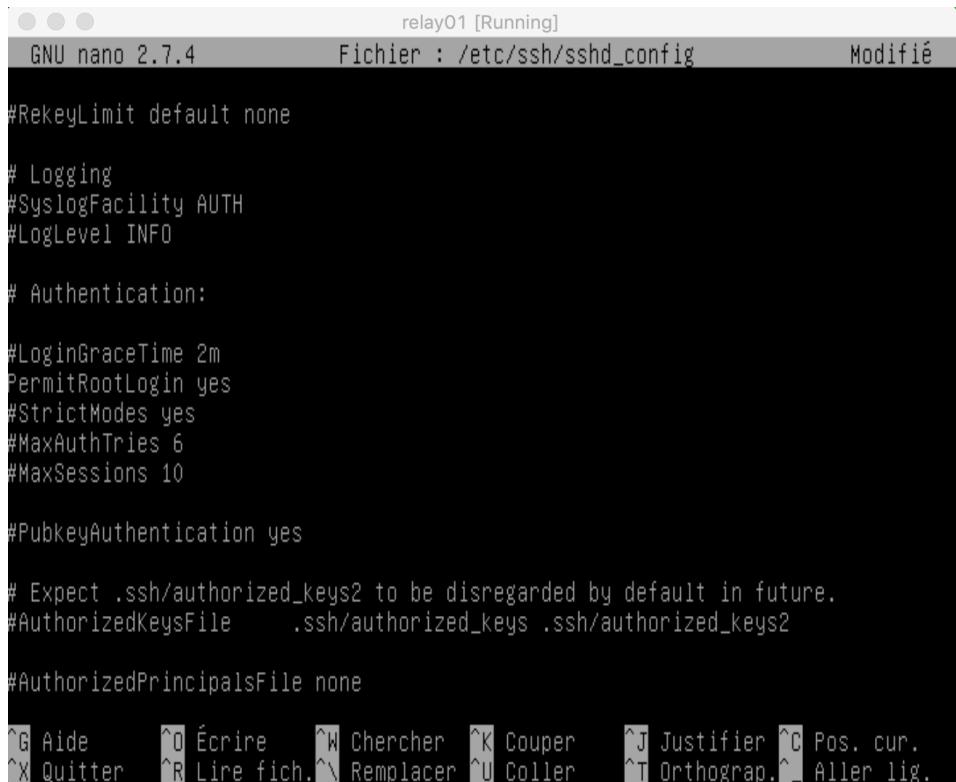
4.3.1. SSH depuis PC client vers SRV WEB et SRV RELAY

On configure dans un premier temps le réseau [LAN](#) pour autoriser le SSH sortant vers le [SRV WEB](#) (172.16.12.1) :

The screenshot shows the 'Edit Firewall Rule' configuration window. The 'Action' dropdown is set to 'Pass'. The 'Disabled' section has a checkbox 'Disable this rule' which is unchecked. The 'Interface' is set to 'LAN'. The 'Address Family' is 'IPv4'. The 'Protocol' is 'TCP'. In the 'Source' section, the 'Source' dropdown is set to 'Single host or alias' with value '172.16.11.20'. The 'Source Port Range' is set to 'SSH (22)' for both 'From' and 'To' fields. In the 'Destination' section, the 'Destination' dropdown is set to 'Single host or alias' with value '172.16.12.1'. The 'Destination Port Range' is set to 'SSH (22)' for both 'From' and 'To' fields.

Ici, le poste [client](#) et lui seul peut accéder en SSH au [serveur WEB](#) situé sur un tout autre réseau. L'inverse n'est pas possible.

Avant, il faut autoriser le login en tant que `root` sur le `serveur web` et sur le `serveur relais`:



The screenshot shows a terminal window titled "relay01 [Running]" with the command "GNU nano 2.7.4". The file being edited is "/etc/ssh/sshd_config". The status bar indicates "Modifié". The configuration file contains several lines of SSH daemon configuration, including settings for rekeying, logging, authentication methods (including root login), and session limits. At the bottom of the screen, there is a menu bar with various keyboard shortcuts for navigating the nano editor.

```
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

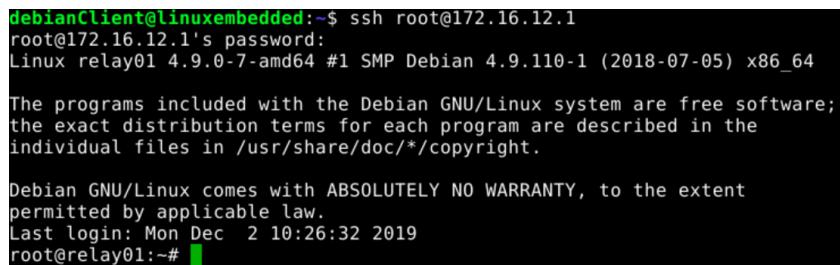
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

^G Aide      ^O Écrire   ^W Chercher  ^K Couper    ^J Justifier ^C Pos. cur.
^X Quitter   ^R Lire fich. ^\ Remplacer ^U Coller    ^T Orthograp.^_ Aller lig.
```

Après un reboot, la connexion SSH se fait sans problème.

L'opération est la même concernant la connexion SSH depuis le poste `client` vers le poste `web`.



The screenshot shows a terminal window with the command "debianClient@linuxembedded:~\$ ssh root@172.16.12.1". The password is entered, and the system responds with the Debian 4.9.110-1 (2018-07-05) x86_64 distribution information. It then displays the standard Debian GNU/Linux copyright notice, which states that the programs are free software and comes with ABSOLUTELY NO WARRANTY. The last login information is shown as "Last login: Mon Dec 2 10:26:32 2019" followed by a root prompt "root@relay01:~#".

```
debianClient@linuxembedded:~$ ssh root@172.16.12.1
root@172.16.12.1's password:
Linux relay01 4.9.0-7-amd64 #1 SMP Debian 4.9.110-1 (2018-07-05) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 2 10:26:32 2019
root@relay01:~#
```

4.3.2. HTTP, HTTPS depuis le PC client vers le SRV_WEB

Ici, on établit une règle autorisant les flux **HTTP** et **HTTPS** vers le serveur **web** situé en **172.16.13.1**.

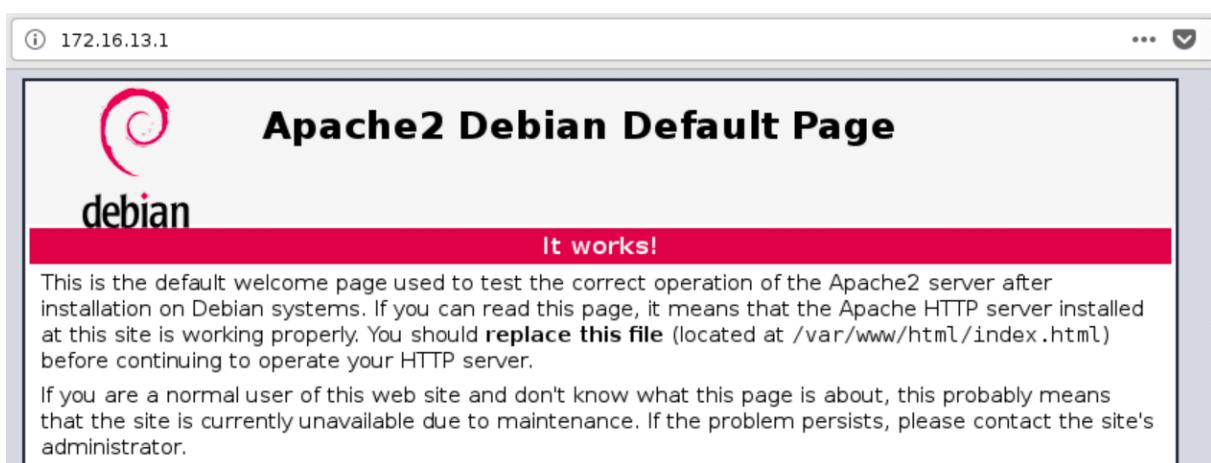
Ainsi, on obtient les deux règles suivantes :

HTTP / HTTPS							
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0 / 0 B	IPv4 TCP	PosteClient	*	ServeurWeb	80 (HTTP)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0 / 0 B	IPv4 TCP	PosteClient	*	ServeurWeb	443 (HTTPS)

Voici, pour exemple, le détail de la règle établie pour **HTTPS** :

The screenshot shows the 'Edit Firewall Rule' dialog. The 'Action' dropdown is set to 'Pass'. Under 'Disabled', there is a checkbox for 'Disable this rule'. The 'Interface' dropdown is set to 'LAN'. The 'Address Family' dropdown is set to 'IPv4'. The 'Protocol' dropdown is set to 'TCP'. In the 'Source' section, the 'Source' dropdown is set to 'Single host or alias' with 'PosteClient' selected. The 'Destination' section shows 'Destination' set to 'Single host or alias' with 'ServeurWeb' selected. Under 'Destination Port Range', 'From' is set to 'HTTPS (443)' and 'To' is set to 'Custom'.

La connexion au serveur web s'effectue sans problème :



4.3.3. HTTP, HTTPS depuis le SRV_Relais vers le SRV_WEB

Ici, on établit une règle autorisant les flux **HTTP** et **HTTPS** depuis le **serveur relais** vers le **serveur web**.

Ainsi on obtient les deux règles suivantes :

HTTP / HTTPS vers SRV_Web									
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0 / 0 B	IPv4 TCP	ServeurRelais	*	ServeurWeb	80 (HTTP)	*	none
					*	ServeurWeb	443 (HTTPS)	*	none

Voici, pour exemple, le détail de la règle établie pour **HTTPS** :

Firewall / Rules / Edit

Edit Firewall Rule

Action: Pass

Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled: Disable this rule
Set this option to disable this rule without removing it from the list.

Interface: VLANDMZRELAIS
Choose the interface from which packets must come to match this rule.

Address Family: IPv4
Select the Internet Protocol version this rule applies to.

Protocol: TCP
Choose which IP protocol this rule should match.

Source

Source: Invert match. Single host or alias: ServeurRelais /

Display Advanced
The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.

Destination

Destination: Invert match. Single host or alias: ServeurWeb /

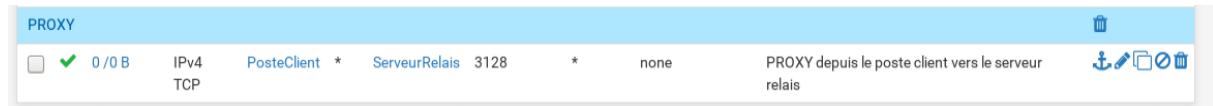
Destination Port Range: HTTPS (443) From: Custom To: Custom
Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

4.3.4. PROXY depuis le PC Client vers SRV_Relais

Le principe ici est d'utiliser le serveur relais comme proxy pour le poste client. Ainsi, il est nécessaire d'établir une règle permettant d'utiliser le serveur relais depuis le client.

Le [proxy](#) que l'on va utiliser est le [proxy Squid](#) qui est open-source.

Le port applicatif de Squid est le port [3128](#), il faut donc établir une règle autour de ce port là en port destination, pour le poste client et pour le serveur relais :



Ici, la configuration détaillée pour le poste client :

A detailed screenshot of a firewall rule configuration. The interface has several tabs at the top: Firewall / Rules / Edit. The main area is titled 'Edit Firewall Rule'.

- Action:** Set to 'Pass'. A note says: 'Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.'
- Disabled:** An unchecked checkbox labeled 'Disable this rule'. A note says: 'Set this option to disable this rule without removing it from the list.'
- Interface:** Set to 'LAN'. A note says: 'Choose the interface from which packets must come to match this rule.'
- Address Family:** Set to 'IPv4'. A note says: 'Select the Internet Protocol version this rule applies to.'
- Protocol:** Set to 'TCP'. A note says: 'Choose which IP protocol this rule should match.'
- Source:** A section with a 'Source' dropdown set to 'PosteClient'. It includes an 'Invert match.' checkbox and a 'Display Advanced' button. A note says: 'The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.'
- Destination:** A section with a 'Destination' dropdown set to 'ServeurRelais'. It includes an 'Invert match.' checkbox and a 'Display Advanced' button. A note says: 'Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.'

On peut tester le bon fonctionnement de la règle en essayant d'accéder au serveur proxy via le navigateur, en renseignant le port applicatif de [squid](#) :

ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: [L](#)

Invalid URL

Some aspect of the requested URL is incorrect.

Some possible problems are:

- Missing or incorrect access protocol (should be "http://" or similar)
- Missing hostname
- Illegal double-escape in the URL-Path
- Illegal character in hostname; underscores are not allowed.

Your cache administrator is [webmaster](#).

Generated Tue, 03 Dec 2019 22:14:30 GMT by relay01 (squid/3.5.23)

On remarque (une erreur, certes) que la réponse de notre requête est une réponse de la part de l'applicatif [squid](#). Le proxy est donc correctement installé.

4.3.5. PROXY depuis le SRV_Web vers SRV_Relais

Le principe ici est d'utiliser le serveur relais comme [proxy](#) pour le serveur web. Ainsi, il est nécessaire d'établir une règle permettant d'utiliser le serveur relais depuis le serveur web.

Le proxy que l'on va utiliser est le proxy [Squid](#) qui est open-source.

Le port applicatif de [Squid](#) est le port [3128](#), il faut donc établir une règle autour de ce port là en port destination, pour le serveur web et pour le serveur relais :



Ici, la configuration détaillée pour le serveur web :

A screenshot of a detailed firewall rule configuration. The rule is titled "Edit Firewall Rule".

- Action:** Pass (selected). Hint: Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
- Disabled:** Disable this rule. Set this option to disable this rule without removing it from the list.
- Interface:** VLANDMZWEB. Choose the interface from which packets must come to match this rule.
- Address Family:** IPv4. Select the Internet Protocol version this rule applies to.
- Protocol:** TCP. Choose which IP protocol this rule should match.
- Source:** **Source:** Invert match. Single host or alias: ServeurWeb / . **Display Advanced:** The Source Port Range for a connection is typically random and almost never equal to the destination port. In most cases this setting must remain at its default value, any.
- Destination:** **Destination:** Invert match. Single host or alias: ServeurRelais / .
Destination Port Range: From (other) 3128 To (other) 3128. Custom. Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

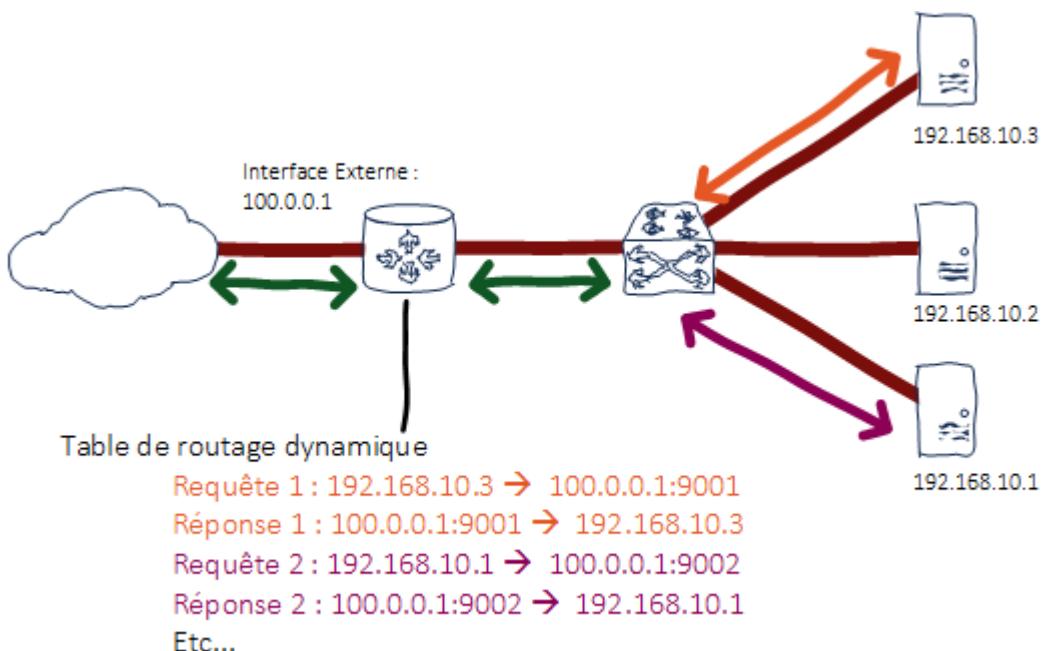
4.4. Règles de NAT

Pour rappel, il existe deux types de NAT :

- Le NAT statique : cela correspond à une translation d'adresse statique. On effectue généralement une translation entre une IP dite publique (externe) et une IP dite privée (interne) pour tout (ou partie) des ports sur lesquels arrivent les paquets sur l'interface publique. Ainsi, c'est grâce à ce procédé qu'un serveur WEB en local dans un réseau peut être accessible depuis Internet.
- Le NAT dynamique : il s'agit de la situation inverse, il permet de mettre un ensemble de machine derrière une ou plusieurs IPs publiques. C'est donc via ce procédé qu'une machine sur un réseau local peut accéder à Internet, utilisant l'adresse IP publique de l'hôte avec qui la translation a été effectuée.

L'exemple ci-dessous a été tiré du site it-connect.fr et illustre bien le propos :

« Si les trois postes décident d'aller sur internet. Le routeur va enregistrer que l'IP interne 192.168.10.3 va être translatée en 100.0.0.1:9001 (par exemple), le second échange sera translaté en 100.0.0.1:9002 pour les ports sources côté internet. Ainsi, quand la réponse d'internet reviendra sur le port 100.0.0.1:9001, le routeur saura qu'il faut renvoyer ces paquets vers 192.168.10.3. C'est une affectation, qu'elle soit par port ou par IP+port, qui est dynamique et éphémère car générée sur demande jusqu'à la fin d'un échange de paquet ou d'une connexion : »

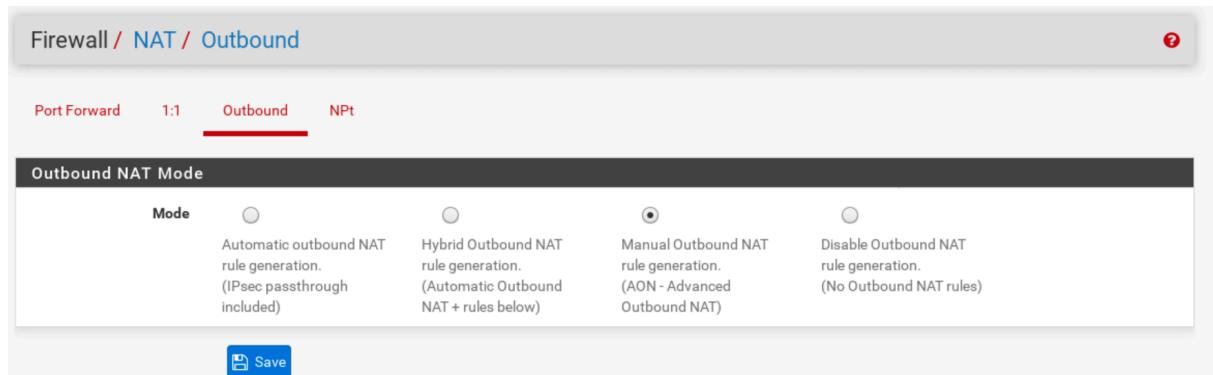


Ces procédés compris, on passe au paramétrage du côté de notre pare-feu.

4.4.1. Flux sortants : NAT dynamique en utilisant l'@IP de l'interface WAN pour le trafic sortant depuis le LAN, la DMZ RELAY et la DMZ WEB

L'objectif de cette partie est de créer un [NAT dynamique](#) permettant à des flux en provenance du réseau interne de pouvoir sortir sur le [WAN](#).

Pour cela, il nous suffit de configurer l'onglet [Outbound](#) sous [NAT](#) mode en mode [Manual Outbound](#). Des règles par défaut sont appliqués, permettant les flux sortants depuis le [LAN](#) vers le [WAN](#).



Maintenant, l'ensemble du trafic sortant depuis le [LAN](#), la [DMZ RELAIS](#) et la [DMZ WEB](#) est autorisé sur le [WAN](#).

Ici, un test avec un ping vers l'adresse IP de Google depuis le poste client :

```
debianClient@linuxembedded:~/Bureau/keypax-0.02b$ ping 216.58.213.163
PING 216.58.213.163 (216.58.213.163) 56(84) bytes of data.
64 bytes from 216.58.213.163: icmp_seq=1 ttl=50 time=95.0 ms
64 bytes from 216.58.213.163: icmp_seq=2 ttl=50 time=93.6 ms
64 bytes from 216.58.213.163: icmp_seq=3 ttl=50 time=74.6 ms
64 bytes from 216.58.213.163: icmp_seq=4 ttl=50 time=93.7 ms
```

Et un autre avec un host :

```
debianClient@linuxembedded:/var/log$ host 8.8.8.8
8.8.8.8.in-addr.arpa domain name pointer dns.google.
```

4.4.2. Flux entrants : NAT statique pour le HTTP, HTTPS, SMTP vers SRV RELAY

On filtre maintenant les paquets en provenance du [WAN](#) vers le serveur relais qui récupère l'ensemble des trames [HTTP](#) / [HTTPS](#) et [SMTP](#).

Un [alias](#) pour les ports a été créé et paramétré comme suit :

The screenshot shows a web-based management interface for a firewall. The top navigation bar has tabs: IP, Ports (which is highlighted with a red underline), URLs, and All. Below the navigation is a sub-header "Firewall Aliases Ports". A table lists a single alias entry:

Name	Values
HTTPS_HTTP_SMBT	80, 443, 25



Edit Redirect Entry

Disabled	<input type="checkbox"/> Disable this rule
No RDR (NOT)	<input type="checkbox"/> Disable redirection for traffic matching this rule This option is rarely needed. Don't use this without thorough knowledge of the implications.
Interface	WAN
Choose which interface this rule applies to. In most cases "WAN" is specified.	
Protocol	TCP
Choose which protocol this rule should match. In most cases "TCP" is specified.	
Source	Display Advanced
Destination	<input type="checkbox"/> Invert match. WAN address Type Address/mask
Destination port range	Other From port HTTPS_HTTP_SMT Other To port HTTPS_HTTP_SMT Custom
Specify the port or port range for the destination of the packet for this mapping. The 'to' field may be left empty if only mapping a single port.	
Redirect target IP	ServeurRelais
Enter the internal IP address of the server on which to map the ports. e.g.: 192.168.1.12	
Redirect target port	Other Port HTTPS_HTTP_SMT Custom
Specify the port on the machine with the IP address entered above. In case of a port range, specify the beginning port of the range (the end port will be calculated automatically).	

4.5. Matrice des flux autorisés finale

Ci-dessous l'export des différents flux autorisés sur notre réseau final :

The screenshot displays two panels of network rules configuration, likely from a MikroTik RouterOS interface.

Panel 1 (Top): WAN Rules

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0 / 1 Kib	*	Reserved Not assigned by IANA	*	*	*	*	*	*	Block bogon networks	
0 / 0 B	IPv4 TCP	WAN net	*	172.16.12.1	443 (HTTPS)	*	none			
0 / 0 B	IPv4 TCP	WAN net	*	172.16.12.1	80 (HTTP)	*	none			
0 / 0 B	IPv4 TCP	WAN net	*	PosteClient	53 (DNS)	*	none		Réponse DNS depuis le net vers le client	
0 / 0 B	IPv4 *	WAN net	*	*	*	*	*	none		

Panel 2 (Bottom): LAN Rules

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0 / 1.11 MiB	*	*	*	LAN Address	80	*	*	*	Anti-Lockout Rule	
SSH										
0 / 4 Kib	IPv4 TCP	PosteClient	*	ServeurRelais	22 (SSH)	*	none		SSH depuis le PC client vers le SRV Relais	
0 / 0 B	IPv4 TCP	PosteClient	*	ServeurWeb	22 (SSH)	*	none		SSH depuis le PC client vers le SRV Web	
Default rules for any										
0 / 0 B	IPv4 *	PosteClient	*	*	*	*	none		Default allow LAN to any rule	
0 / 0 B	IPv6 *	LAN net	*	*	*	*	none		Default allow LAN IPv6 to any rule	
HTTP / HTTPS										
0 / 0 B	IPv4 TCP	PosteClient	*	ServeurWeb	80 (HTTP)	*	none			
0 / 0 B	IPv4 TCP	PosteClient	*	ServeurWeb	443 (HTTPS)	*	none			
0 / 0 B	IPv4 TCP/UDP	PosteClient	*	*	53 (DNS)	*	none			
PROXY										
0 / 14.15 MiB	IPv4 TCP/UDP	PosteClient	*	ServeurRelais	3128	*	none		PROXY depuis le poste client vers le serveur relais	
ICMP										
0 / 0 B	IPv4 ICMP any	PosteClient	*	*	*	*	none			

Firewall / Rules / VLANDMZRELAIS

Floating WAN LAN VLANDMZRELAIS **VLANDMZWEB**

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0 /0 B	IPv4 TCP	ServeurRelais	*	*	25 (SMTP)	*	none		Anchor Edit Copy Delete	
<input type="checkbox"/>	✓ 0 /284 B	IPv4 TCP/UDP	ServeurRelais	*	*	53 (DNS)	*	none		Anchor Edit Copy Delete	
HTTP / HTTPS vers WAN NET											
<input type="checkbox"/>	✓ 0 /6.14 MiB	IPv4 TCP	ServeurRelais	*	*	443 (HTTPS)	*	none		Anchor Edit Copy Delete	
<input type="checkbox"/>	✓ 94 /2.80 MiB	IPv4 TCP	ServeurRelais	*	*	80 (HTTP)	*	none		Anchor Edit Copy Delete	
<input type="checkbox"/>	✓ 0 /2 KiB	IPv4 ICMP any		*	*	*	*	none		Anchor Edit Copy Delete	
HTTP / HTTPS vers SRV_Web											
<input type="checkbox"/>	✓ 0 /887 KIB	IPv4 TCP	ServeurRelais	*	ServeurWeb	80 (HTTP)	*	none		Anchor Edit Copy Delete	
<input type="checkbox"/>	✓ 0 /0 B	IPv4 TCP	ServeurRelais	*	ServeurWeb	443 (HTTPS)	*	none		Anchor Edit Copy Delete	

Firewall / Rules / VLANDMZWEB

Floating WAN LAN VLANDMZRELAIS **VLANDMZWEB**

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0 /0 B	IPv4 TCP	VLANDMZWEB net	*	*	53 (DNS)	*	none		Anchor Edit Copy Delete	
<input type="checkbox"/>	✓ 0 /0 B	IPv4 ICMP any		*	*	*	*	none		Anchor Edit Copy Delete	
PROXY											
<input type="checkbox"/>	✓ 0 /0 B	IPv4 TCP	ServeurWeb	*	ServeurRelais	3128	*	none	PROXY depuis le serveur web vers le serveur relais	Anchor Edit Copy Delete	

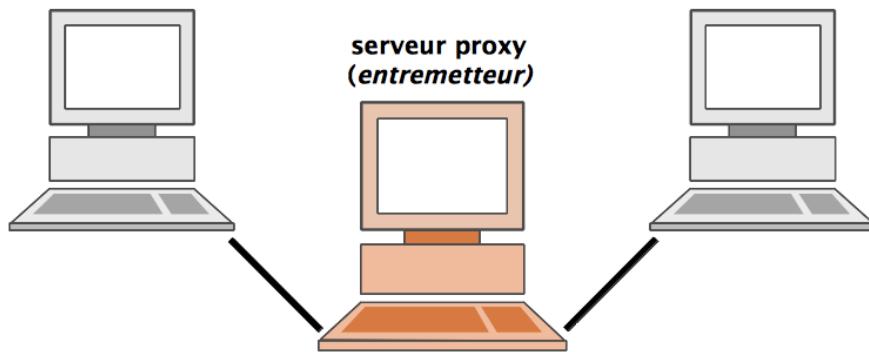
Ainsi que les détails des alias utilisés :

Firewall Aliases IP

Name	Values	Description
PosteClient	172.16.11.20	
ServeurRelais	172.16.12.1	
ServeurWeb	172.16.13.1	
VLAN1pfSense	172.16.11.254	
WANout	192.168.1.25	pfSense cote WAN

4.6. Configuration du proxy

Cette partie détaille la configuration de **SQUID** sur le serveur relais. **Squid** est un proxy qui joue le rôle d'intermédiaire entre deux hôtes. Dans notre cas, il joue l'intermédiaire entre le réseau local et le [WAN](#).



4.6.1. Configuration Squid sur SRV RELAY

Le proxy permet de faire le lien entre les flux [HTTP](#), [HTTPS](#) et [SNMP](#) entrants (depuis le [WAN](#)) et sortants (depuis le [LAN](#)).

Le fichier de configuration du proxy se trouve dans [/etc/squid/squid.conf](#).

Il est recommandé de partir avec une configuration propre de [squid](#).

On configure le proxy en partant du modèle [squid.conf](#) suivant :

```
##### ACL DEFINITION #####
# Définition des différentes zones de notre réseau

acl lan src 172.16.11.0/24
acl web src 172.16.13.0/24

http_access allow lan
http_access allow web

# Définition des ports utilisés par le réseau et redirige vers le relais
acl SSL_ports port 443
acl Safe_ports port 80    # HTTP
acl Safe_ports port 443 # HTTPS
acl Safe_ports port 25    # SMTP
acl Safe_ports port 53    # DNS

# Désactiver tous les protocoles sauf les ports sûres
http_access deny !Safe_ports
```

```
# Désactiver l'accès pour tous les réseaux sauf les clients de l'ACL Lan
# deny = refuser ; ! = sauf ; lan = nom de l'ACL à laquelle on fait référence.
http_access deny !lan

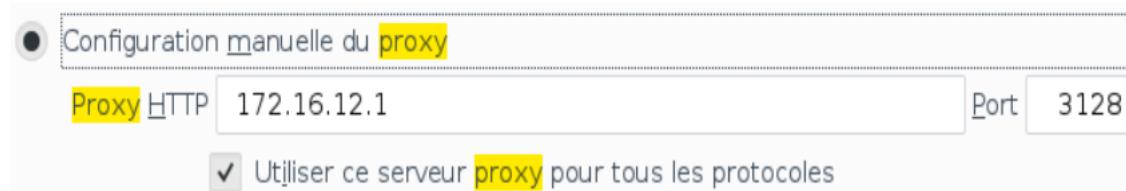
# Port à utiliser
http_port 3128
```

Une fois les règles définies, on enregistre et on restart le service `squid.service`.

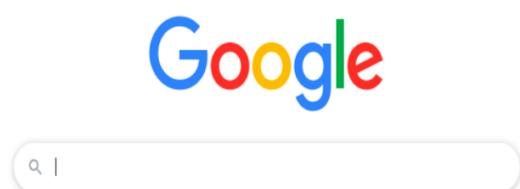
```
root@relay01:~# service squid stop
root@relay01:~# service squid start
```

```
root@relay01:~# service squid status
● squid.service - LSB: Squid HTTP Proxy version 3.x
  Loaded: loaded (/etc/init.d/squid; generated; vendor preset: enabled)
  Active: active (running) since Mon 2019-12-02 13:24:16 CET; 32s ago
    Docs: man:systemd-sysv-generator(8)
 Process: 689 ExecStop=/etc/init.d/squid stop (code=exited, status=0/SUCCESS)
 Process: 757 ExecStart=/etc/init.d/squid start (code=exited, status=0/SUCCESS)
 Main PID: 797 (squid)
   Tasks: 4 (limit: 4915)
  CGroup: /system.slice/squid.service
          ├─795 /usr/sbin/squid -YC -f /etc/squid/squid.conf
          ├─797 (squid-1) -YC -f /etc/squid/squid.conf
          ├─798 (logfile-daemon) /var/log/squid/access.log
          └─799 (pinger)
```

Il est ensuite important de renseigner le proxy depuis le navigateur du poste client :



On teste la connexion et on vérifie le bon fonctionnement (en cas de dysfonctionnement, un fichier de `log` par défaut est consultable sous `/var/log/squid/access.log`) :



4.6.2. Configuration Squidguard sur SRV RELAY

Le fichier de configuration de [squidguard](#) se trouve dans [/etc/squidguard/squidGuard.conf](#)

On commence par réaliser un back-up du fichier de configuration de [squidguard](#) :

```
$ cp /etc/squidguard/squidGuard.conf /etc/squidguard/squidGuard.back
```

On peut désormais passer à la partie configuration, étape par étape :

```
#  
# CONFIG FILE FOR SQUIDGUARD  
  
#  
  
dbhome /var/lib/squidguard/db  
logdir /var/log/squid  
  
# Postes sources autorisés  
src admin {  
    ip 172.16.11.20visu
```

On définit dans cette première partie les paramètres de base du fichier de configuration [squidguard](#). On renseigne ainsi le chemin d'accès pour le fichier de log que va générer [squidGuard](#)

Ensuite, on passe à la définition des différentes sources. On considère que dans notre LAN se trouve un poste [admin](#) parmi d'autres clients, et les règles lui seront appliquées directement sur son IP.

Ainsi, l'adresse IP du poste admin étant statique dans notre configuration réseau, on crée un groupe admin dans ce fichier de configuration de [squidguard](#).

Ce groupe nous permettra de raffiner les droits plus tard.

On passe ensuite aux règles de filtrage :

```
dest reseaux {  
    domainlist reseaux/domains  
    urllist reseaux/urls  
}
```

Ici, [squidguard](#) va venir filtrer l'ensemble des domaines et urls recensés dans le dossier [/var/lib/squidguard/db/reseaux](#) et [/var/lib/squidguard/db/eshop](#).

Protips : La génération de la base de données nécessite le lancement de la commande ci-dessous :

```
root@relay01:~# squidGuard -C all
```

Ci-dessous un exemple d'arborescence avec le dossier `eshop`. Le fichier `urls` contient l'ensemble des urls bloquées idem pour le fichier `domains` qui contient les noms de domaines interdits (par exemple, `facebook.com`) :

```
root@relay01:/# cd /var/lib/squidguard/db/reseaux/
root@relay01:/var/lib/squidguard/db/reseaux# ls
domains domains.db urls urls.db
root@relay01:/var/lib/squidguard/db/reseaux# cat domains
facebook.com
root@relay01:/var/lib/squidguard/db/reseaux#
```

Ensuite, on associe les groupes créés avec les interdictions précédemment définies :

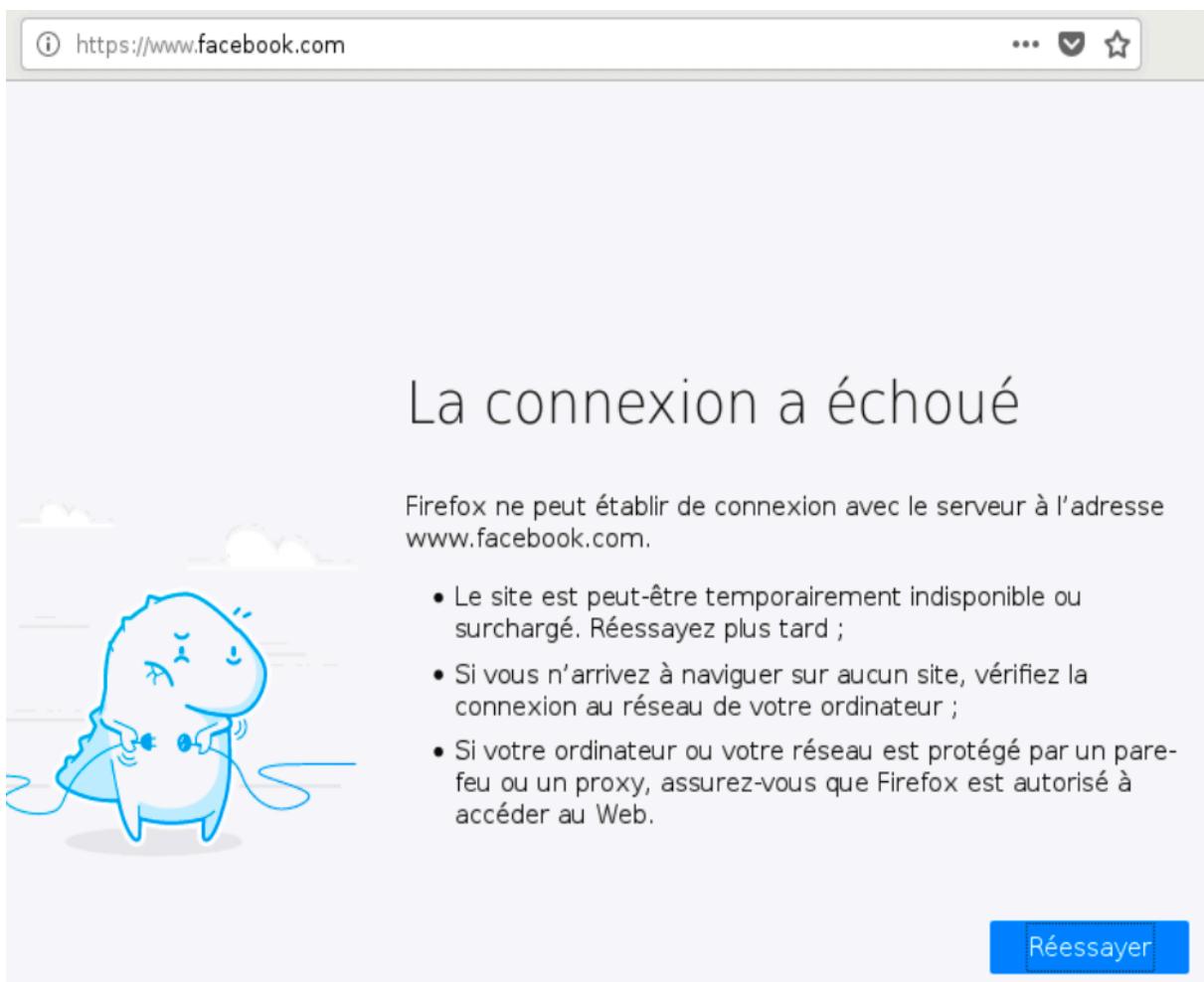
```
#La règle avec les interdictions
acl {
    default {
        pass !reseaux
    }
}
```

Dans cet exemple, le groupe `default` peut accéder à n'importe quel site sauf `facebook.com`. Ils sont également directement rediriger vers `Google.fr` lorsque ils tentent à ce domaine (on aurait également pu les rediriger vers une page d'alerte générée par `squid`).

Pour rendre le tout fonctionnel, on rectifie les droits sur le dossier (`squid` étant lancé en root, la `database` générée ne sera pas accessible en lecture / écriture par `squid`), et on régénère :

```
root@relay01:~# chown -R squid:squid /var/lib/squidguard/ && systemctl
restart squid.service
```

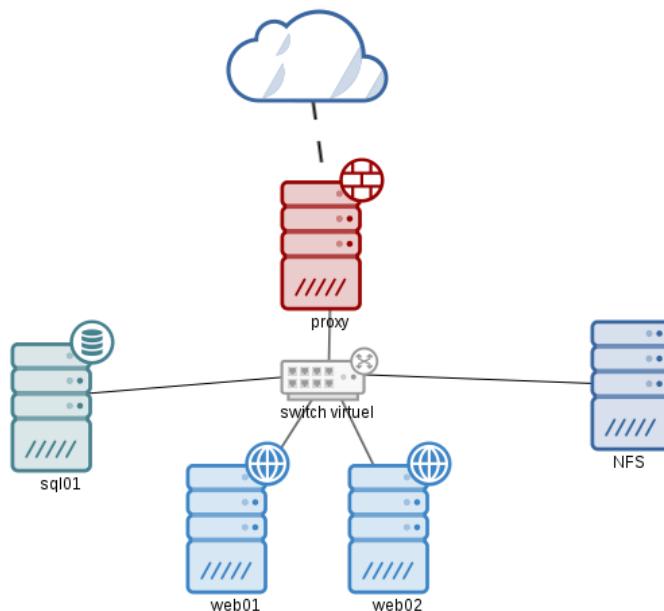
Le proxy est maintenant fonctionnel :



4.7. Configuration du reverse proxy

4.7.1. Configuration HAProxy sur SRV RELAY

On passe maintenant à l'installation et la configuration du [HAProxy](#) sur le serveur relais. Ce reverse proxy va permettre de rendre notre site web accessible depuis Internet. Notre serveur web étant en backend, c'est l'adresse IP du reverse-proxy qui sera visible en front.



Ce schéma est donné à titre informatif, il n'est pas représentatif de l'infrastructure existante.

Par défaut, le service est activé sur le serveur :

```
root@relay01:/etc/squidguard# service haproxy status
● haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset:
  Active: active (running) since Mon 2019-12-02 13:13:14 CET; 1h 19min ago
    Docs: man:haproxy(1)
          file:/usr/share/doc/haproxy/configuration.txt.gz
  Process: 354 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRAOPTS (code=
 Main PID: 366 (haproxy-systemd)
   Tasks: 3 (limit: 4915)
  CGroup: /system.slice/haproxy.service
          └─366 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy.cfg -
              ├─379 /usr/sbin/haproxy-master
              ├─381 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.p
```

Comme pour la configuration de squid, [HAProxy](#) dispose d'un fichier de configuration. Il va de soi qu'on prend toujours la précaution de sauvegarder le fichier de configuration initial :

```
root@relay01:/etc/haproxy# cp haproxy.cfg haproxy.cfg.bak
```

On passe ensuite à la partie configuration du `haproxy.conf` :

```
global
#Paramètres globaux de haproxy, les logs (dans syslog),
#l'utilisateur et le group de fonctionnement du service.

    log 127.0.0.1 local0
    log 127.0.0.1 local1 notice
    user haproxy
    group haproxy


defaults
# mode : mode de fonctionnement par défaut ;
# maxconn : nombre de connexion maximum acceptée sur le frontal, (protection anti
DDOS)
# timeout : différents timeout permettant de couper les connexions trop longues
# errorfile : Définition des pages d'erreur génériques (exemple avec une erreur
404)

    log global
    mode http
    option httplog
    option dontlognull
    retries 3
    option redispatch
    maxconn 2000
    timeout connect 5s
    timeout client 30s
    timeout server 10s
    timeout http-request 5s
    errorfile 404 /etc/haproxy/errors/404.http


listen webstats
#haproxd dispose d'une page de stats accessible. On choisit le port applicatif
(ici 8080).
#On applique ensuite les differentes statistiques que l'on veut voir
    bind 172.16.12.1:8080
    stats enable
    stats hide-version
    stats scope webfarm
    stats scope webservers
    stats uri /
    stats realm Haproxy\ Statistics
    stats auth haproxy:secret
    stats refresh 10s
```

```

#Fonction d'écoute sur le port 80 du serveur relais
listen webfarm
    #On écoute sur le port 80
    bind 172.16.12.1:80
    #Et on fait référence en backend : la partie webserver paramétré sur notre
serveur web.
    default_backend webservers

#Définition du serveur backend : application du reverse proxy sur notre serveur
web.
backend webservers
    mode http
    balance roundrobin
    #les requêtes sont réparties les unes après les autres sur
    #chaque serveur de façon uniforme.

    cookie LBN insert indirect nocache
    #nom du cookie
    #Le proxy va positionner un cookie ici nommé LBN avec pour contenu le nom
du serveur
    #backend ;
    option httpclose
    option forwardfor
    #On renseigne ici les paramètres de notre serveur web en backend.
    server web01 172.16.13.1:80 weight 55 cookie web01 check inter 1s

```

Cette configuration permet donc, en généralisant, d'écouter sur le port `80` du serveur relais et de rediriger les flux vers le serveur `backend` qui est notre serveur web. Ce dernier est ainsi accessible depuis le net en étant masqué par le reverse proxy.

Le reverse proxy est également monitorable via un `webstats` accessible sur le port `8080` du serveur relais.

On restart `haproxy` et la configuration est appliquée :

```

root@relay01:/etc/haproxy# service haproxy restart
root@relay01:/etc/haproxy# service haproxy status
● haproxy.service - HAProxy Load Balancer
  Loaded: loaded  (/lib/systemd/system/haproxy.service; enabled; vendor
  preset: enabled)
  Active: active (running) since Mon 2019-12-02 15:05:47 CET; 5s ago

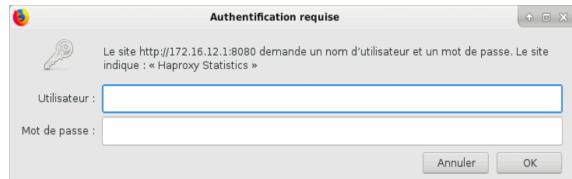
```

```

Docs: man:haproxy(1)
      file:/usr/share/doc/haproxy/configuration.txt.gz
Process: 1149 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRAOPTS
(code=exited, status=0/SUCCESS)

```

Aperçu du [webstats](#) via le port 8080 de notre serveur relais :



On renseigne les [creds](#) (ligne `stats auth haproxy:secret` , et on accède au moniteur :

webfor[1]																
Queue	Session rate			Sessions				Bytes								
	Cur	Max	Limit	Cur	Max	Limit	Total	In	Out	Denied						
Frontend	0	3	-	0	1	2000	3	1261	9800	0	0	0	0	0	0	OPEN
Backend	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	14m12s UP

webserver[1]																	
Queue	Session rate			Sessions				Bytes									
	Cur	Max	Limit	Cur	Max	Limit	Total	In	Out	Denied							
web01	0	0	-	0	3	0	1	3	1	5m09s	1261	9800	0	0	0	0	14m12s UP
Backend	0	0	0	0	3	0	1	200	1	5m09s	1261	9800	0	0	0	0	14m12s UP

On teste la connexion sur le port 80 de notre [serveur relais](#) :

```
http://172.16.12.1:80
```

On est bien rediriger vers le serveur web et la page HTML d'accueil du serveur [apache](#) de notre [webserveur](#) :



Protips : On pourrait tester le bon fonctionnement du pare-feu en rajoutant une machine côté [WAN](#) et en autorisant un flux sur le pare-feu depuis cette IP vers le port [80](#) du pare-feu. A travers une règle de [port forwarding](#), le port [80](#) du pare-feu est redirigé vers le port [80](#) du serveur relais ([reverse proxy](#)) qui lui-même renvoie sur le serveur [back-end](#) sur la [DMZ Web](#).

4.8. UTM PfSense

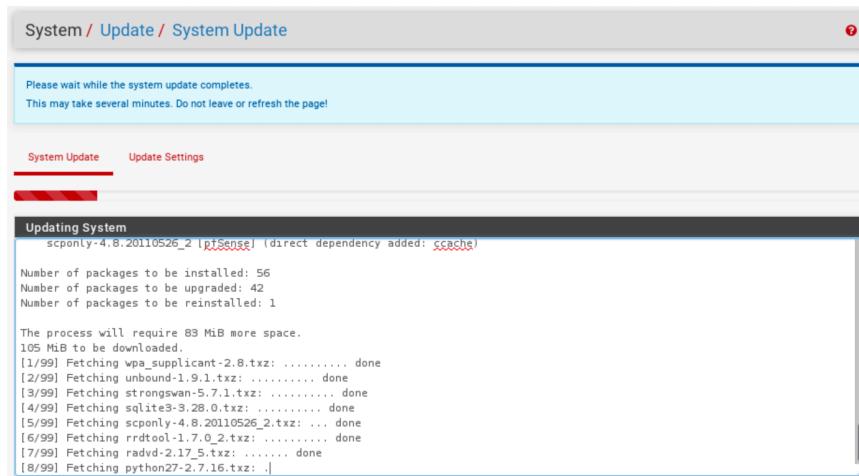
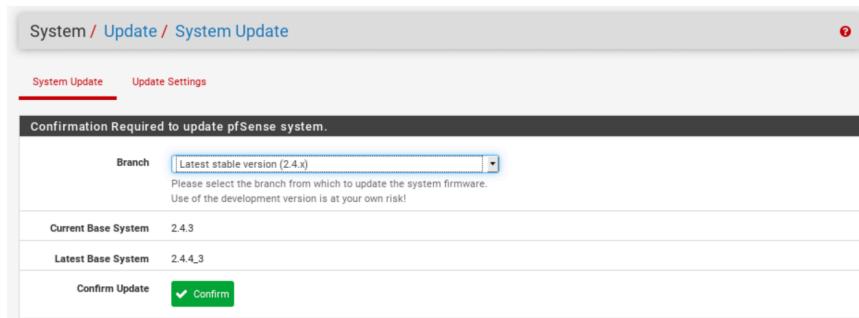
« UNIFIED THREAT MANAGEMENT, OU UTM (EN FRANÇAIS : GESTION UNIFIEE DES MENACES) EST UN TERME INVENTÉ ET UTILISE POUR DECRIRE DES PARE-FEUX RESEAU QUI POSSEDENT DE NOMBREUSES FONCTIONNALITES SUPPLEMENTAIRES QUI NE SONT PAS DISPONIBLES DANS LES PARE-FEU TRADITIONNELS. »

Wikipedia.

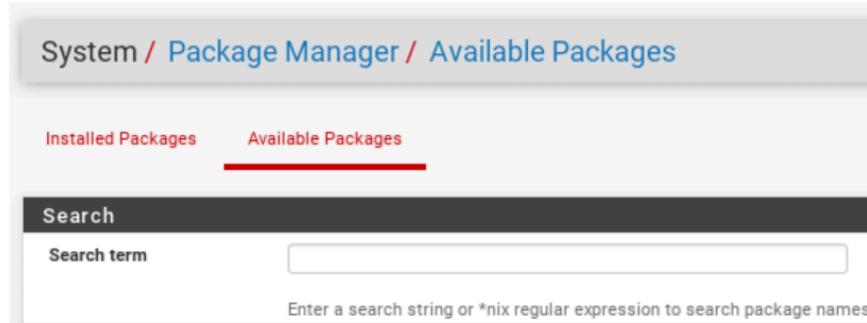
4.8.1. Installer les paquages nécessaires afin d'implémenter les services de proxy et reverse proxy sur le firewall PfSense

Avant de commencer, il convient de vérifier la version du pare-feu et de le mettre à jour si une mise à jour est disponible. L'installation d'un paquet nécessite en effet d'avoir la dernière version du pare-feu.

Pour mettre à jour le pare-feu, on se rend dans l'onglet [System -> Update](#) :



Une fois le pare-feu mis à jour, on se rend dans l'onglet [System -> Package manager](#)

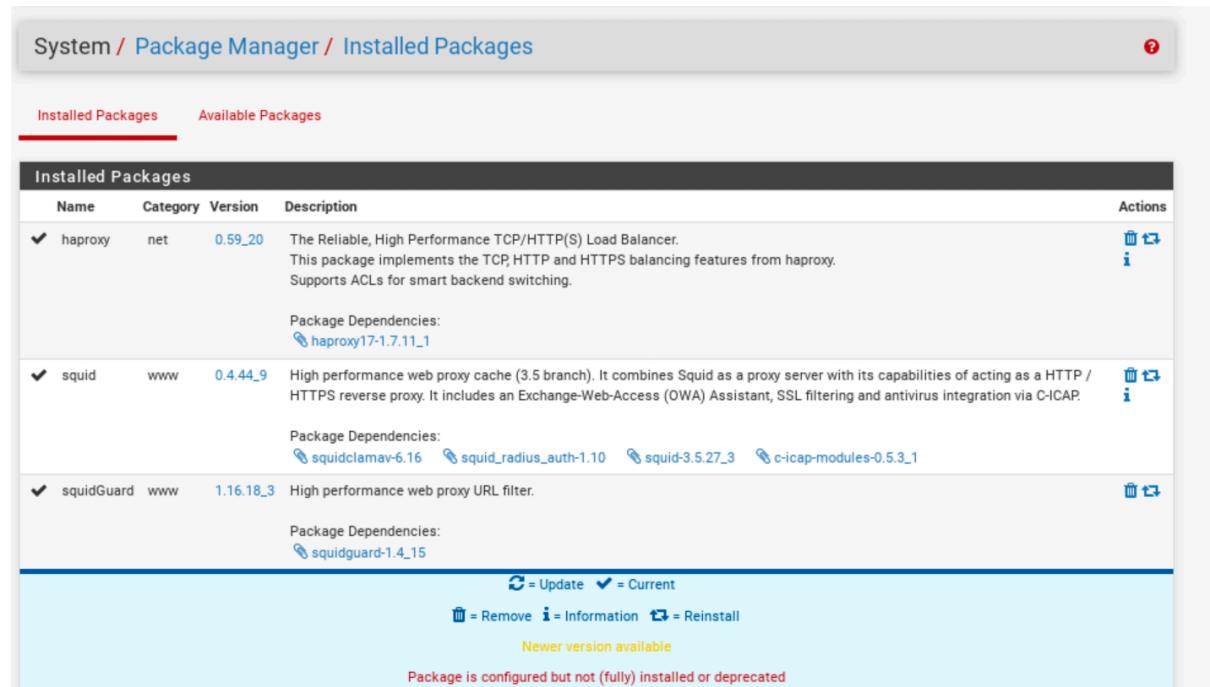


A partir de là, on peut maintenant installer les paquets nécessaires à savoir :

```
squid  
squidGuard  
haproxy
```

L'installation des paquets se fait à travers le bouton [Install](#) situé à droite du paquet à installer.

On vérifie la bonne installation de chaque paquet :

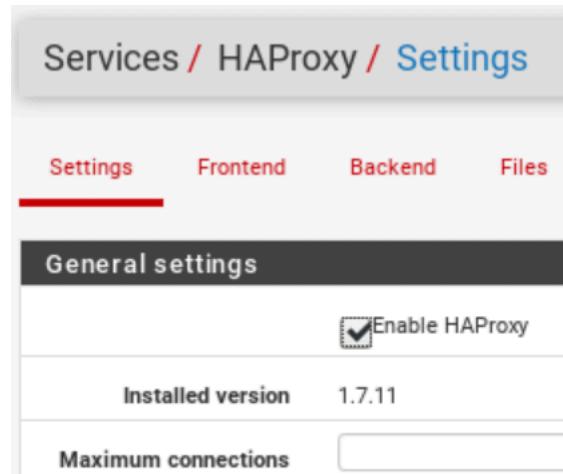


Installed Packages				
Name	Category	Version	Description	Actions
✓ haproxy	net	0.59_20	The Reliable, High Performance TCP/HTTP(S) Load Balancer. This package implements the TCP, HTTP and HTTPS balancing features from haproxy. Supports ACLs for smart backend switching.	Delete Edit Information
Package Dependencies: haproxy17-1.7.11_1				
✓ squid	www	0.4.44_9	High performance web proxy cache (3.5 branch). It combines Squid as a proxy server with its capabilities of acting as a HTTP / HTTPS reverse proxy. It includes an Exchange-Web-Access (OWA) Assistant, SSL filtering and antivirus integration via C-ICAP.	Delete Edit Information
Package Dependencies: squidclamav-6.16 squid_radius_auth-1.10 squid-3.5.27_3 c-icap-modules-0.5.3_1				
✓ squidGuard	www	1.16.18_3	High performance web proxy URL filter.	Delete Edit
Package Dependencies: squidguard-1.4_15				

4.8.2. Configurer des services sur le firewall

- **Haproxy**

On commence par activer le service depuis la page de configuration de [Haproxy](#) :



A partir de là, on peut définir [Haproxy](#) à l'instar de la configuration effectuée depuis le serveur relais. En effet, l'interface graphique dispose de toutes les fonctions configurables sur le [haproxy.conf](#). Il est alors largement possible de retrouver la même configuration que celle effectuée sur le serveur relais. L'interface graphique propose également de l'aide sur chacun des paramètres afin de guider l'utilisateur vers l'utilisation optimale et recommandée du proxy.

Name	<input type="text"/>														
Description	<input type="text"/>														
Status	Active														
External address	Define what ip:port combinations to listen on for incoming connections. Table: <table border="1"><thead><tr><th></th><th>Listen address</th><th>Custom address</th><th>Port</th><th>SSL Offloading</th><th>Advanced</th><th>A</th></tr></thead><tbody><tr><td><input type="checkbox"/> WAN address (IPv4)</td><td><input type="text"/></td><td><input type="text"/></td><td>80</td><td><input type="checkbox"/></td><td><input type="text"/></td><td></td></tr></tbody></table> <p>NOTE: You must add a firewall rules permitting access to the listen ports above. If you want this rule to apply to another IP address than the IP address of the interface chosen above, select it here (you need to define Virtual IP addresses on the first). Also note that if you are trying to redirect connections on the LAN select the "any" option. In the port to listen to, if you want to specify multiple ports, separate them with a comma (.). EXAMPLE: 80,8000 Or to listen on both 80 and 443 create 2 rows in the table where for the 443 you would likely want to check the SSL-offloading checkbox.</p>		Listen address	Custom address	Port	SSL Offloading	Advanced	A	<input type="checkbox"/> WAN address (IPv4)	<input type="text"/>	<input type="text"/>	80	<input type="checkbox"/>	<input type="text"/>	
	Listen address	Custom address	Port	SSL Offloading	Advanced	A									
<input type="checkbox"/> WAN address (IPv4)	<input type="text"/>	<input type="text"/>	80	<input type="checkbox"/>	<input type="text"/>										
Max connections	<input type="text"/>														
Type	<input type="text"/> http / https(offloading)														

- **Squid**

On commence par activer le service depuis la page de configuration de [Squid](#) :

The screenshot shows the 'General' tab selected in the top navigation bar. Below it, the 'Squid General Settings' section is displayed. A checkbox labeled 'Enable Squid Proxy' is checked. A note below it states: 'Important: If unchecked, ALL Squid services will be disabled and stopped.'

Concernant [Squid](#), le principe est le même que pour [Haproxy](#), on retrouve l'intégralité des possibilités sur un fichier [squid.conf](#) à travers l'interface graphique depuis le [pfSense](#), avec l'aide aux utilisateurs sur les recommandations :

This screenshot shows the full 'Squid General Settings' configuration page. It includes sections for 'Proxy Interface(s)' (set to LAN), 'Proxy Port' (set to 3128), 'ICP Port' (left blank), 'Allow Users on Interface' (checked), 'Patch Captive Portal' (disabled), 'Resolve DNS IPv4 First' (unchecked), 'Disable ICMP' (unchecked), and 'Use Alternate DNS Servers for the Proxy Server' (empty input field). A note at the bottom says: 'To use DNS servers other than those configured in System > General Setup, enter the IP(s) here. Separate entries by semi-colons (;)'.

4.8.3. Installer les paquages nécessaires afin d'implémenter un service IPS/IDS

Pour cette partie, la technologie d'IPS/IDS utilisée sera la technologie [Snort](#) :



System / Package Manager / Available Packages

Installed Packages Available Packages

Search term: snort Both Search Clear

Enter a search string or *nix regular expression to search package names and descriptions.

Packages

Name	Version	Description	Action
snort	3.2.9.10	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection.	+ Install

Package Dependencies:

- snort-2.9.15
- barnyard2-1.13_1

4.8.4. Configurer l'IPS/IDS

On passe maintenant à l'installation et la configuration de [Snort](#) :

Services / Snort / Interfaces

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Interface Settings Overview

Interface	Snort Status	Pattern Match	Blocking	Barnyard2 Status	Description	Actions
						+ Add

5. Retour d'expérience

Concernant mon avis personnel sur le projet celui-ci s'est avéré assez intéressant à traiter. Le fait de sécuriser une infrastructure en utilisant du matériel courant en entreprise et ce, depuis la configuration initiale jusqu'à la mise en place finale de la solution m'a apporté beaucoup de connaissances sur des technologiques que je ne connaissais pas (squid, pfSense, haproxy).

Malheureusement, j'ai rencontré énormément de problème entre mes machines virtuelles, de la configuration des interfaces jusqu'au réglage fin des routes et des passerelles en passant par les différents VLANS, j'ai perdu énormément de temps à mettre en place l'architecture initiale.

J'ai également eu un certains problèmes avec l'interface WAN NET qui, sans que je m'en rende compte au premier abord, ne fonctionnait tout simplement pas. Par conséquent j'étais obligé d'ouvrir des flux volontaires abaissant la sécurité de l'infrastructure.

Une fois le tout fonctionnel, c'est assez plaisant d'avoir une architecture qui tourne plutôt bien et un proxy paramétrable à la volée. Encore une fois, beaucoup de concessions en matière de cybersécurité ont dû être faites pour essayer de faire fonctionner l'ensemble.

Le principe du moindre privilège possible n'a pas pu être tenu même si des efforts ont été faits pour limiter les flux aux minimums tout en garantissant un fonctionnement minimal de l'infrastructure.

Assez dommage de ne pas avoir eu des conseils sur les bonnes pratiques de configuration de pfSense dès le départ (alias, restrictions, règles en destination) ; la documentation étant très complète, un petit coup de pouce sur certains points aurait permis de gagner du temps précieux et de pousser le TP un peu plus loin (cf. IDS / IPS / VPN) au lieu d'être bloquer sur une ridicule variable pendant des heures.

Mis à part ce point, le contenu était très intéressant et enrichissant.

Merci pour votre lecture.

Kévin Moreau.

6. Annexes et bibliographies

Ci-dessous les sources et documentations utiles qui ont été utilisées et étudiées dans le cadre de la rédaction de ce rapport.

- *Installer un proxy Squid et un filtrage avec SquidGuard sous Debian | memo-linux.com.* <https://memo-linux.com/installer-un-proxy-squid-et-un-filtrage-avec-squidguard-sous-debian/>. Consulté le 2 décembre 2019.
- kikinovak. « Filtrer le web avec SquidGuard sous CentOS 7 ». *MicroLinux*, 6 avril 2019, <https://www.microlinux.fr/squidguard-centos-7/>.
- *Mise en place d'un serveur Haproxy | Commandes et Système | IT-Connect*. <https://www.it-connect.fr/mise-en-place-dun-serveur-haproxy/>. Consulté le 2 décembre 2019.
- <https://www.symantec.com/content/dam/symantec/docs/data-sheets/web-application-firewall-en.pdf>
- « Qu'est-ce qu'un reverse-proxy ? Le serveur reverse-proxy ». *IONOS Digitalguide*, <https://www.ionos.fr/digitalguide/serveur/know-how/quest-ce-quun-reverse-proxy-le-serveur-reverse-proxy/>. Consulté le 2 décembre 2019.
- *Web Application Firewall 101 | Connect - Edition Diamond*. <https://connect.ed-diamond.com/MISC/MISC-088/Web-Application-Firewall-101>. Consulté le 2 décembre 2019.
- « [SOLVED] Setup Manual Outbound NAT - Section in pfsense docs unclear to me ». *Netgate Forum*, 31 août 2017, <https://forum.netgate.com/topic/119782/solved-setup-manual-outbound-nat-section-in-pfsense-docs-unclear-to-me/2>.
- *Network Address Translation — 1:1 NAT | pfSense Documentation*. <https://docs.netgate.com/pfsense/en/latest/nat/1-1-nat.html>. Consulté le 2 décembre 2019.
- Guillaume. « Mise en place haproxy Debian 8 ». *Aukfood*, <https://www.aukfood.fr/mise-en-place-haproxy-debian-8/>. Consulté le 2 décembre 2019.
- « 12.3. Configuration Squid Red Hat Enterprise Linux 7 ». *Red Hat Customer Portal*, https://access.redhat.com/documentation/fr-fr/red_hat_enterprise_linux/7/html/networking_guide/sec-squid-configuration. Consulté le 4 décembre 2019.

- *SquidGuard[L'internet rapide et permanent]*. http://irp.naint.net/doku.php/220squid:060_squidguard. Consulté le 4 décembre 2019.
- Unknown. « Techs Tricks: Test a HTTP request with wget ». *Techs Tricks*, 11 décembre 2008, <http://techs-tricks.blogspot.com/2008/12/test-http-request-with-wget.html>.
- *Configurer Firefox pour utiliser un proxy HTTP*. <https://www.free-proxy.fr/index.php?menu=firefox-proxy>. Consulté le 4 décembre 2019.
- *Problème:Squid + HTTPS par Anthurus - OpenClassrooms*. <https://openclassrooms.com/forum/sujet/probleme-squid-https>. Consulté le 4 décembre 2019.
- *SquidFaq/SquidLogs - Squid Web Proxy Wiki*. <https://wiki.squid-cache.org/SquidFaq/SquidLogs>. Consulté le 4 décembre 2019.
- *TAG_NONE/409 CONNECT - Squid 3.5.20*. https://www.linuxquestions.org/questions/linux-server-73/tag_none-409-connect-squid-3-5-20-a-4175620518/. Consulté le 4 décembre 2019.