Documentation de l’installation du réseau d'une grande entreprise en devenir.

Procédure rédigée par : Vassenet-Guihot Romain

Table des matières

I - Présentation Générale des Besoins de L'entreprise :

Afin de répondre au mieux à la croissance d'une grande entreprise, la reconfiguration complète du réseau doit être repensé pour servir aux mieux les besoins actuels et futur. L'entreprise est constitué de 5 départements, le service RH, le service comptabilité, le service commercial, les équipes opérationnelles et l'équipe informatique. Voici le cacher des charges à réaliser :

#1 Définir le plan d’adressage réseau

#2 Choisir les différents matériels réseaux et systèmes adaptés

#3 Créer le réseau de l'entreprise en utilisant des VLAN pour chaque service

#4 Configurer les différents équipements réseau

#5 Vérifier le bon fonctionnement du réseau

#6 Installer les services DHCP, DNS, NTP sur le serveur interne

#7 Installer un serveur web sur le serveur DMZ et créer une entrée DNS à usage interne pour accéder à ce site

#8 Stocker des fichiers statiques sur Amazon S3

#9 Installer un proxy Squid par lequel doivent nécessairement passer les employés pour aller sur Internet.

#10 Vérifier que ce proxy permet un gain de performances pour l’accès aux pages les plus couramment consultées.

#11 Sécuriser le réseau avec des pare-feux

II - Définir le Plan d'Adressage Réseau de L'entreprise :

Nous devons dans un premier temps établir un plan d’adressage de notre réseau afin de définir une architecture la plus près possible des effectifs. Pour cela nous avons découpé les plages réseaux grâce

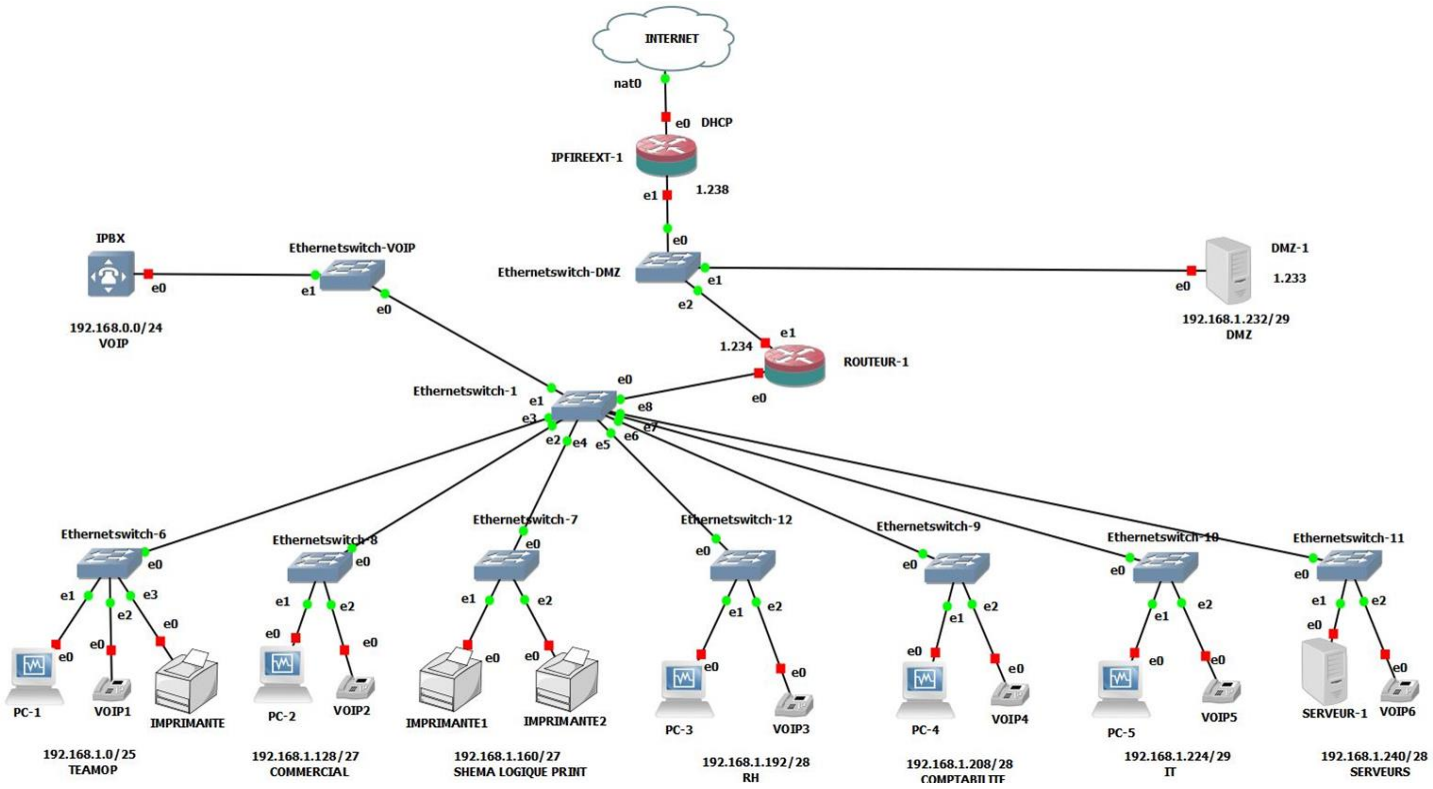
aux masque de sous réseau afin d'obtenir le bon nombre d'adresses IP pour chaque départements. Il faudra également prendre en compte le doublement des effectifs d'ici 5 ans. Afin de faciliter le découpage, j'ai commencé de façon décroissante afin de « tailler » au plus près les besoins d'adresses IP pour chaque département.

Departement / Zone	Adresse Réseau	Masque de Sous réseau	1er IP	Derniere IP	Broadcast	Nombre IP possible	Nombre machines du département (x2 d'ici 5 ans)
VOIP	192.168.0.0	255.255.255.0 / 24	192.168.0.1	192.168.0.254	192.168.0.255	254	148
TEAM OP	192.168.1.0	255.255.255.128 / 25	192.168.1.1	192.168.1.126	192.168.1.127	126	104
COMMERCIAL	192.168.1.128	255.255.255.224 /27	192.168.1.129	192.168.1.158	192.168.1.159	30	24
Imprimantes	192.168.1.160	255.255.255.224 /27	192.168.1.161	192.168.1.190	192.168.1.191	30	4+++
RH	192.168.1.192	255.255.255.240 / 28	192.168.1.193	192.168.1.206	192.168.1.207	14	10
Comptabilté	192.168.1.208	255.255.255.240 / 28	192.168.1.209	192.168.1.222	192.168.1.223	14	6
IT	192.168.1.224	255.255.255.248 /29	192.168.1.225	192.168.1.230	192.168.1.231	6	4
DMZ	192.168.1.232	255.255.255.248 /29	192.168.1.233	192.168.1.238	192.168.1.239	6	2
Serveurs	192.168.1.240	255.255.255.240 /28	192.168.1.241	192.168.1.254	192.168.1.255	14	2+++

III - Virtualisation Général GNS3 de l'Architecture Réseau :

Notre plan d'adressage réseau définit, nous devons désormais virtualiser notre architecture GNS3. Nous utiliserons virtualbox pour la virtualisation des machines virtuelles dans un soucis de compatibilité des VLAN sous GNS3. Il faudra également s'assurer de cocher la case sur chaque machine virtuel gns3 « allow gns3 to use any configured virtualbox adapter ». Chaque machine virtuel sera sous une distribution Debian 9, sauf pour Ipfire qui est basé sous LFS. Nous aurons un switch coeur LAN, tout ses ports excepté e1 seront en Trunk de manière à faire passer plusieurs VLAN par port du switch. A l'exception du port e0 du switch VOIP qui est en access tout les port e0 des switch LAN seront en Trunk car VLAN10 VOIP, VLAN40 imprimante sur chaque switch de chaque département + son VLAN PC associé : VLAN20 = TeamOP, VLAN30 = Commercial, VLAN50 = RH, VLAN 60 =Compta, VLAN70 = IT

Pour rappel les VLAN permette de segmenter de façon logique le réseau. Les réseau local virtuel permette également de répondre à des besoins de sécurité afin d'isoler certaine partie du réseau sans recours aux règles de routage d'un routeur. Permet également l'optimisation du matériel ou encore de réserver de la bande passante sur un réseau définit.



IV - Configuration Firewall Externe Ipfire:

Après avoir définit les bases de notre architecture sous GNS3, Plan d'adressage, Installation des VM sous VirtualBox et configuration de nos switch avec les bon VLAN, nous devons commencer à configurer le matériel indispensable au bon fonctionnement de notre réseau. Ici IpFire nous servira de pare-feu externe. Afin de configurer ce pare-feu opensource de distribution LFS, il nous faudra définir le nombre d'interface à utiliser et la configuration du protocole internet version 4. Au premier lancement de IpFire, la configuration nous propose de choisir le nombre d'interface voulu, ici nous en configurerons deux.

L'interface nommé Green0 faisant office de passerelle du réseau DMZ aura comme configuration manuelle :

IP : 192.168.1.238

Masque de sous réseau : 255.255.255.248

L'interface WAN Red0 faisant office de « porte de sortie vers l'extérieur » récupérera une adresse IP automatiquement grâce au DHCP pointant dans notre exemple de virtualisation et non de réalité d'entreprise sur une appliance GNS3 NAT. En réalité, nous aurions plusieurs interfaces WAN relié à

différents modem de ligne ADSL/VDSL ou encore un transeiver sur un port WAN pour la Fibre Optique.

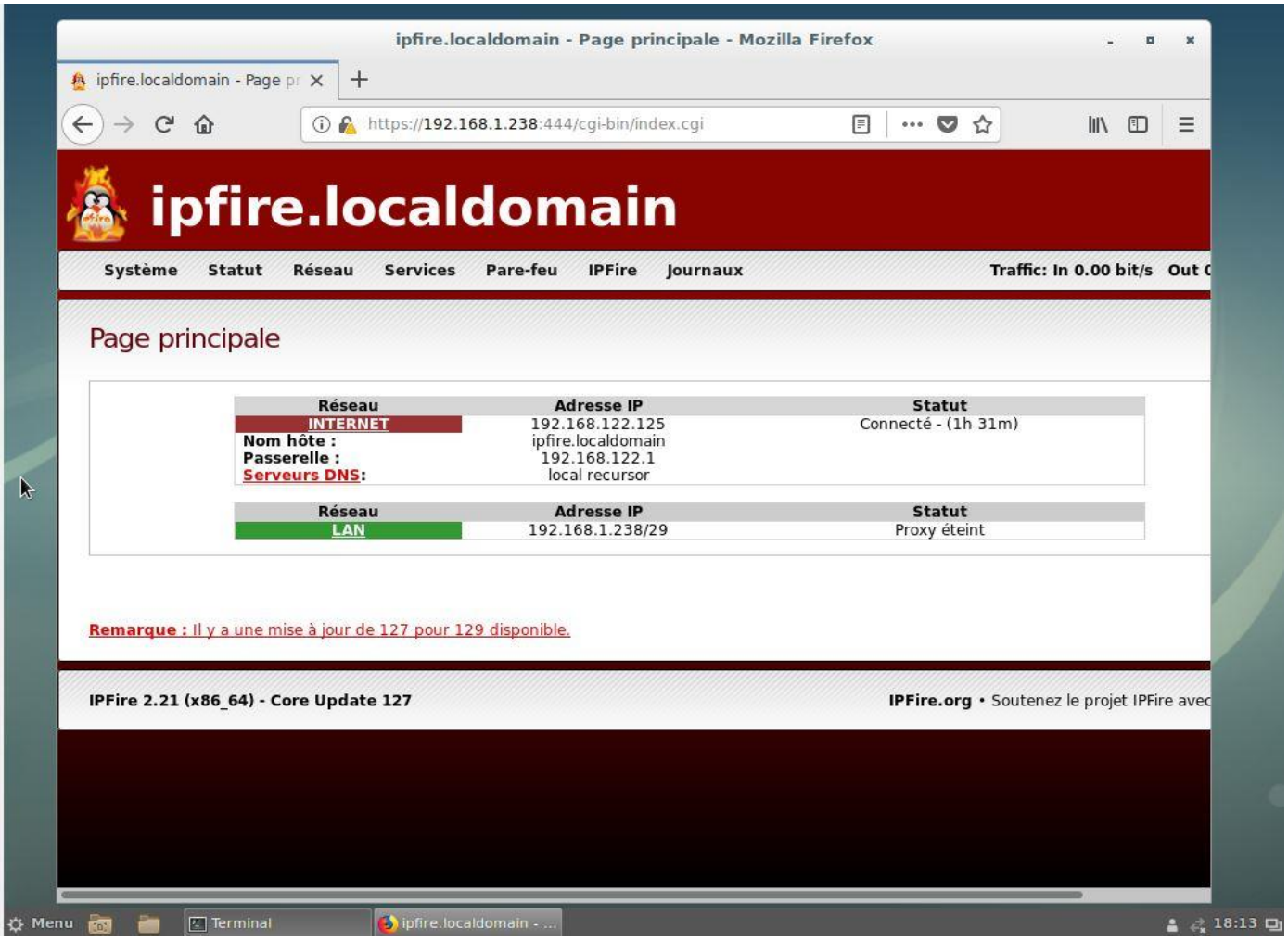
Afin d'accéder au dashboard de IpFire, nous allons nous connecter sur via le serveur DMZ sur sa passerelle DMZ Green0 en web.

https://192.168.1.238:444

Lorsque nous avions essayé de nous connecter malheureusement ça ne fonctionnait pas car IpFire n'était pas en écoute sur les PORT HTTP/HTTPS

Pour le voir netstat -laputen | grep LISTEN

Une fois ce problème résolu, nous pouvons interagir en web avec la partie graphique de Ipfire.



Nous allons activer le DNNSEC de manière à éviter certains problèmes de sécurités liés au protocole DNS comme le Spoofing DNS.

Nous aurons également une configuration standard du firewall de manière à éviter les attaques DDOS

Nous appliquerons la restriction unique pour sortir vers internet par le Proxy Squid sur notre PareFeu Interne avec iptables car la DMZ, d'ou ça définition est isolé de notre réseau Interne. Notre LAN discutera avec le serveur Interne pour passer par le proxy Squid.

Afin de vérifier que la résolution de nom depuis notre Parefeu Ipfire fonctionne, nous allons rendre immutable le fichier resolv.conf :

rm -f /etc/resolv.conf

editor /etc/resolv.conf

On y ajoute alors notre serveur dns validé par DNNSEC: nameserver 8.8.8.8

chattr +i /etc/resolv.conf (enlever le mode immutable avec chattr -i / ainsi après chaque redémarrage le fichier sera persistant et nom modifiable)

V - Configuration Du Serveur Interne :

Le serveur interne est sans aucun doute le point majeur des différents services réseaux interne à l'entreprise. En effet celui-ci permettra de répondre à différents besoins tel que la sécurité avec le proxy Squid, l'attribution automatique des IP pour chaque département tout en faisant le liens avec leur VLAN associés (DHCP), Le serveur DNS Bind9 avec une entrée à usage interne pour accéder au serveur web de la DMZ. Sans oublier le service NTP afin de synchroniser les machines du réseau interne en interrogeant un serveur de temps.

V.1 - Configuration VLAN & DHCP:

Il faut comme lors du précédent projet installer les packets pour faire un serveur DHCP mais il faut également installer les packets pour configurer les VLAN.

apt-get install isc-dhcp-server

apt-get install vlan

Afin de propager plusieurs VLAN sur un même lien physique, il faudra configurer un trunk avec la norme établie dot1q

pour cela nous allons déclarer cette norme en y ajoutant 8021q :

nano /etc/modules

Il faut également vérifier que le paramètre suivant est bien ajouté dans chaque interfaces virtuel VLAN :

vlan-raw-device [Interface]

Voici la configuration réseau des interfaces virtuels VLAN configuré sur le serveur interne via le fichier de configuration /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

#auto enp0s3
#iface enp0s3 inet static
#address 192.168.1.252
#netmask 255.255.255.240

auto enp0s3.10
iface enp0s3.10 inet static
address 192.168.0.1
netmask         255.255.255.0
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8
```

```
auto enp0s3.20
iface enp0s3.20 inet static
address 192.168.1.1
netmask 255.255.255.128
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8
gateway 192.168.1.126

auto enp0s3.30
iface enp0s3.30 inet static
address 192.168.1.129
netmask 255.255.255.224
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8

auto enp0s3.40
iface enp0s3.40 inet static
address 192.168.1.161
netmask 255.255.255.224
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8

auto enp0s3.50
iface enp0s3.50 inet static
address 192.168.1.193
netmask 255.255.255.240
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8
```

auto enp0s3.60

iface enp0s3.60 inet static

address 192.168.1.209

netmask 255.255.255.240

vlan-raw-device enp0s3

dns-nameservers 8.8.8.8

auto enp0s3.70

iface enp0s3.70 inet static

address 192.168.1.225

netmask 255.255.255.248

vlan-raw-device enp0s3

dns-nameservers 8.8.8.8

Nous allons ensuite configurer le service DHCP en veillant à respecter que pour chaque réseau la première et la dernière adresse IP disponible de la plage ne soit pas inclus dans le range DHCP. Ainsi nous pourrons définir dans la configuration DHCP pour chaque réseau la passerelle qui sera l'interface virtuel VLAN associé soit la première adresse IP ainsi que les DNS via la dernière adresse IP de la plage réseau associé.

Fichier de configuration dhcpd.conf

dhcpd.conf

Sample configuration file for ISC dhcpd

#

option definitions common to all supported networks...

option domain-name "mondomaine.com";

#option domain-name-servers 8.8.8.8, 8.8.4.4;

default-lease-time 600;

max-lease-time 7200;

```
subnet 192.168.0.0 netmask 255.255.255.0 {
        range 192.168.0.2 192.168.0.253;
        option routers 192.168.0.254;
}

subnet 192.168.1.0 netmask 255.255.255.128 {
        range 192.168.1.2 192.168.1.125;
        option routers  192.168.1.126;
        option domain-name-servers 192.168.1.1;
}

subnet 192.168.1.128 netmask 255.255.255.224 {
        range 192.168.1.130 192.168.1.157;
        option routers 192.168.1.158;
        option domain-name-servers 192.168.1.129;

}

subnet 192.168.1.160 netmask 255.255.255.224 {
        range 192.168.1.162 192.168.1.189;
        option routers 192.168.1.190;
        option domain-name-servers 192.168.1.161;

}

subnet 192.168.1.192 netmask 255.255.255.240 {
        range 192.168.1.194 192.168.1.205;
        option routers 192.168.1.206;
        option domain-name-servers 192.168.1.193;

}
```

```
subnet 192.168.1.208 netmask 255.255.255.240 {
        range 192.168.1.210 192.168.1.221;
        option routers 192.168.1.222;
        option domain-name-servers 192.168.1.209;

}

subnet 192.168.1.224 netmask 255.255.255.248 {
        range 192.168.1.226 192.168.1.229;
        option routers 192.168.1.230;
        option domain-name-servers 192.168.1.225;

}

subnet 192.168.1.232 netmask 255.255.255.248 {
        range 192.168.1.234 192.168.1.237;
        option routers 192.168.1.238;
}

subnet 192.168.1.240 netmask 255.255.255.240 {
        range 192.168.1.242 192.168.1.253;
        option routers 192.168.1.254;
}
```

Il faut également penser à rajouter son interface.VLANX dans INTERFACESV4 via le nano /etc/default/isc-dhcp-server :

```
INTERFACESv4="enp0s3.10 enp0s3.20 enp0s3.30 enps0s3.40 enp0s3.50 enp0s3.60 enp0s3.70 enp0s3.80"
```

V.2 - Configuration Service DNS, Proxy SQUID, NTP :

Afin d'installer un serveur DNS qui fera office de résolution de nom par l'ensemble des machines sur le réseau interne, il faut installer bind9

apt-get install bind9

Après quoi nous devons configurer le fichier named.conf.options. Il faudra y ajouter dans les forwarders les dns principaux tel que google et cloudflare.

Il faudra également accepter la restriction des acl définit, ici nous l'avons nommé internals, instruction donné pour la configuration de allow-query

internals a été définit dans le fichier principal named.conf chargeant l'ensemble des sous configurations bind9.

allow-query-cache permet également une rapidité dans la requête des nom de domaine déjà recherché.

allow-recursion permet de réaliser des requêtes récursive sur le réseau définit

fichier de configuration named.conf.options

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

         forwarders {
                8.8.8.8;
                8.8.4.4;
```

```
                1.1.1.1;
        };

        //========================================================================
========
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
========
        dnssec-validation no;
        dnssec-enable no;

        auth-nxdomain no;    # conform to RFC1035
        //listen-on-v6 { any; };

        // Accepter les requêtes pour le réseau interne uniquement
        allow-query { internals; };
        allow-recursion { internals; };
        allow-query-cache { internals; };

};
```

Il nous faut également créer un entrée DNS à usage interne afin d'accéder au serveur web installé sur le serveur de DMZ.

Pour cela nous allons créer deux fichier de configuration portant le nom de notre domaine

db.mondomaine.com et db.mondomaine.com.inv (pour la résolution inversé car on demandera au serveur DNS de renvoyer le nom de domaine pleinement qualifié à partir de l'adresse ip de la machine qui le demande.

Dans notre fichier db.mondomaine.com, on définit le nom du serveur, l'adresse IP du serveur web et son alias.

Fichier de configuration db.mondomaine.com

```
$TTL    86400
@ IN SOA webdmz.mondomaine.com. root.mondomaine.com.  (
        201903111  ; Serial  -> N° de série à incrémenter à chaque modif
        28800      ;Refresh ->  A l'expiration du délai Refresh exprimé en
        14400      ; Retry
        3600000    ; Expire
        86400 )    ; Minimum -> Durée de vie minimum du cache en secondes

;** Les 3 lignes suivantes permettent au serveur de se retrouver lui même
            NS      webdmz.mondomaine.com.                  ;Nom du serveur
webdmz          IN      A       192.168.1.233               ;Adresse IP du serveur web
www             IN      CNAME   webdmz                      ;Alias
```

Fichier de configuration db.mondomaine.com.inv

```
$TTL    86400
@ IN SOA webdmz.mondomaine.com. root.mondomaine.com.  (
        201903111  ; Serial  -> N° de série à incrémenter à chaque modif
        28800      ;Refresh ->  A l'expiration du délai Refresh exprimé en
        14400      ; Retry
        3600000    ; Expire
        86400 )    ; Minimum -> Durée de vie minimum du cache en secondes

;** Les 3 lignes suivantes permettent au serveur de se retrouver lui même
            NS      webdmz.mondomaine.com.                  ;Nom du serveur
233             PTR     webdmz.mondomaine.com.
```

Il faut également ajouter dans le fichier de configuration général de bind9 ces deux fichiers afin que ces fichiers « zones » soit chargé au lancement du serveur DNS

Fichier de configuration général bind9 named.conf

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local
// Gérer les acls
acl internals {
192.168.1.0/25;
192.168.1.128/27;
192.168.1.160/27;
192.168.1.192/28;
192.168.1.208/28;
192.168.1.224/29;
#192.168.1.240/28;
#127.0.0.1/32;
 };
include "/etc/bind/named.conf.options";
#include "/etc/bind/named.conf.local";
#include "/etc/bind/named.conf.default-zones";

zone "mondomaine.com" {
        type master;
        file "/etc/bind/db.mondomaine.com";
        forwarders{};
```

};

zone "1.168.192.in-addr.arpa" {

type master;

file "/etc/bind/db.mondomaine.com.inv";

forwarders{};

};

Enfin, après avoir relancé le service bind9, nous vérifions que tout fonctionne en faisant /etc/init.d/bind9 status

```
root@debian9:/etc/bind# /etc/init.d/bind9 status
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2019-04-14 19:09:29 CEST; 1h 28min ago
     Docs: man:named(8)
 Main PID: 861 (named)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/bind9.service
           └─861 /usr/sbin/named -f -u bind
```

Maintenant que le serveur DNS bind9 est configuré pour usage du serveur web et qu'il utilise uniquement ses forwarders et non le resolv.conf pour résolution de nom(/etc/default/bind9 : RESOLVCONF=NO nous allons installer le proxy squid :

apt-get install squid

On doit définir les acl et l'entrée DNS

Fichier de configuration squid.conf

acl localnet src 192.168.1.0/24

acl SSL_ports port 443

acl Safe_ports port 80 # http

acl Safe_ports port 21 # ftp

acl Safe_ports port 443 # https

acl Safe_ports port 70 # gopher

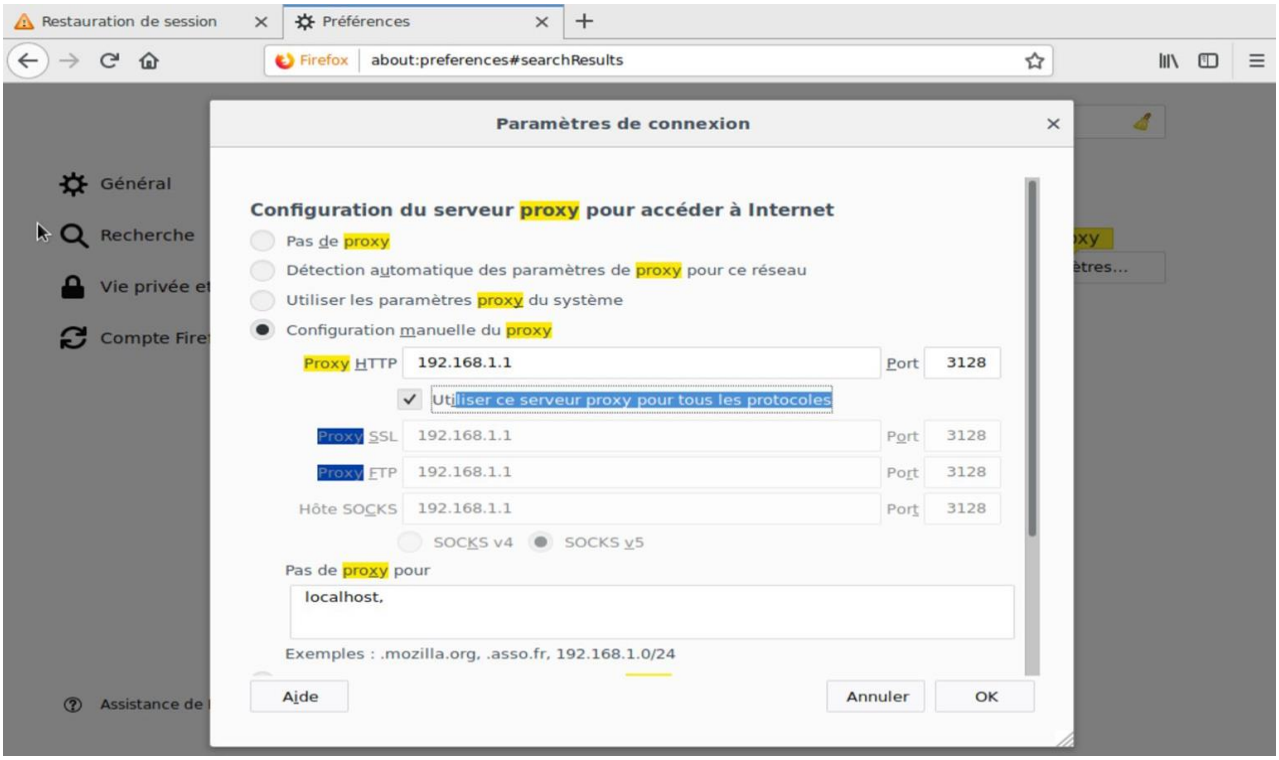
```
acl Safe_ports port 210		# wais
acl Safe_ports port 1025-65535	# unregistered ports
acl Safe_ports port 280		# http-mgmt
acl Safe_ports port 488		# gss-http
acl Safe_ports port 591		# filemaker
acl Safe_ports port 777		# multiling http
acl CONNECT method CONNECT
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
http_access allow localnet
http_access allow localhost
http_access deny all
http_port 3128
coredump_dir /var/spool/squid
refresh_pattern ^ftp:		1440	20%	10080
refresh_pattern ^gopher:	1440	0%	1440
refresh_pattern -i (/cgi-bin/|\?) 0	0%	0
refresh_pattern .		0	20%	4320
dns_nameservers 192.168.1.1
```

On vérifie également que squid fonctionne correctement avec /etc/init.d/squid status

```
root@debian9:/etc# init.d/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 Sun 2019-04-14 19:09:30 CEST; 2h 9min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 869 ExecStart=/etc/init.d/squid start (code=exited, status=0/SUCCESS)
 Main PID: 982 (squid)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/squid.service
           ├─ 980 /usr/sbin/squid -YC -f /etc/squid/squid.conf
           ├─ 982 (squid-1) -YC -f /etc/squid/squid.conf
           ├─ 983 (logfile-daemon) /var/log/squid/access.log
           └─1498 (pinger)

avril 14 19:09:30 debian9 systemd[1]: Starting LSB: Squid HTTP Proxy version 3.x...
avril 14 19:09:30 debian9 squid[869]: Starting Squid HTTP Proxy: squid.
avril 14 19:09:30 debian9 systemd[1]: squid.service: PID file /var/run/squid.pid not readable (yet?) aft…rectory
avril 14 19:09:30 debian9 squid[980]: Squid Parent: will start 1 kids
avril 14 19:09:30 debian9 squid[980]: Squid Parent: (squid-1) process 982 started
avril 14 19:09:30 debian9 systemd[1]: squid.service: Supervising process 982 which is not our child. We'… exits.
avril 14 19:09:30 debian9 systemd[1]: Started LSB: Squid HTTP Proxy version 3.x.
```

Plus qu'a configurer notre proxy sur firefox via l'adresse ip de l'interface du VLAN associé sur le réseau auquel on appartient sur le port squid par défault 3128. Par exemple si l'on se trouve sur le réseau TeamOP :



Désormais nous devons installer et configurer le serveur NTP.

apt-get install ntp

Nous devons ensuite configurer le serveur ntp en y ajoutant les serveur de temps pour la zone française : nano /etc/ntp.conf – 0.fr.pool.ntp.org [...]

Fichier de configuration ntp.conf

/etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

Enable this if you want statistics to be logged.

#statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats

filegen loopstats file loopstats type day enable

filegen peerstats file peerstats type day enable

filegen clockstats file clockstats type day enable

You do need to talk to an NTP server or two (or three).

#server ntp.your-provider.example

pool.ntp.org maps to about 1000 low-stratum NTP servers. Your server will

pick a different set every time it starts up. Please consider joining the

pool: <http://www.pool.ntp.org/join.html>

pool 0.fr.pool.ntp.org iburst

pool 1.fr.pool.ntp.org iburst

pool 2.fr.pool.ntp.org iburst

pool 3.fr.pool.ntp.org iburst

Pour synchroniser une machine sur le serveur temp il faudra simplement faire ntpdate 192.168.1.X (X=Interface selon sur lequel VLAN on se trouve)

VI - Configuration du Serveur Web DMZ & Amazon S3

Sur le serveur DMZ nous devons configurer l'interface réseau manuellement avec l'adresse que nous avions choisit pour notre serveur Web :

192.168.1.233/29 et la passerelle qui est notre IpFire en 192.168.1.238

Dans un soucis de performance sur le serveur Web qui aura une base de donnée Mysql ou MariaDB très importante, nous décidons d'installer Nginx pour répondre a un gros trafic. Contrairement à LAMP (Linux, Apache, Mysql, Php) Nginx est plus difficile à configurer car il faut interconnecter chaque service web à Nginx mais la puissance du moteur permettra de gérer le nombre de processeur alloué, la RAM, la bande passante, etc ..

Afin de faire fonctionner php il faudra penser à installer php7-fpm et y ajouter la configuration pour rendre compatible NGINX avec PHP.

```
server {
#Permet d'écouter sur le port 80 de l'IPv4 de votre serveur
      listen 80;
#Permet d'écouter sur le port 80 de l'IPv6 de votre serveur
      listen [::]:80;

#Vous devez renseigner le nom de domaine de votre site internet
     server_name webdmz.mondomaine.com;

#Définit le répertoire qui va accueillir les fichiers de votre site internet
      root /var/www/monsupersite;

#Permet de définir l'ordre d'exécution de votre index. Ici, s'il y a deux index.php/html à la racine du site, index.php sera exécuté en priorité
      index index.php index.html;

#Ici, on donne l'ordre d'afficher une page 404 sur la totalité du site si un fichier n'existe pas
      location / {
        try_files $uri $uri/ =404;
    }

         location ~ \.php$ {
                  try_files $uri =404;
                  fastcgi_pass localhost:9000;
                  fastcgi_index index.php;
                  fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
                  include fastcgi_params;
 }
 }
```

Plus qu'a mettre notre site dans var/www/monsupersite en y ajoutant le lien de notre fichier à l'intérieur du bucket Amazon S3 (Que nous verrons ensuite)

```
<!DOCTYPE HTML>
<!--
	Aerial by HTML5 UP
	html5up.net | @ajlkn
	Free for personal and commercial use under the CCA 3.0 license (html5up.net/license)
-->
<html>
	<head>
		<title>AIC PORJET4</title>
		<meta charset="utf-8" />
		<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
		<link rel="stylesheet" href="assets/css/main.css" />
		<noscript><link rel="stylesheet" href="assets/css/noscript.css" /></noscript>
	</head>
	<body class="is-preload">
		<div id="wrapper">
			<div id="bg"></div>
			<div id="overlay"></div>
			<div id="main">

				<!-- Header -->
					<header id="header">
						<h1>VASSENET-GUIHOT ROMAIN</h1>
						<p>Administrateur Infrastructure & Cloud  &bull;  Administrateur Infrastructure & Cloud   &bull;  Openclassrooms <br/> <a href="https://s3.eu-west-3.amazonaws.com/guihotbucket/linux_logo.jpg">MOn Stoackage AMAZON S3</a><p>
						<nav>
```

```
                                    <ul>
                                        <li><a href="#" class="icon fa-twitter"><span class="label">Twitter</span></a></li>
                                        <li><a href="#" class="icon fa-facebook"><span class="label">Facebook</span></a></li>
                                        <li><a href="#" class="icon fa-dribbble"><span class="label">Dribbble</span></a></li>
                                        <li><a href="#" class="icon fa-github"><span class="label">Github</span></a></li>
                                        <li><a href="#" class="icon fa-envelope-o"><span class="label">Email</span></a></li>
                                    </ul>
                                </nav>
                            </header>

                        <!-- Footer -->
                            <footer id="footer">
                                <span class="copyright">&copy; Untitled. Design: <a href="http://html5up.net">HTML5 UP</a>.</span>
                            </footer>

                    </div>
                </div>
                <script>
                    window.onload = function() { document.body.classList.remove('is-preload'); }
                    window.ontouchmove = function() { return false; }
                    window.onorientationchange = function() { document.body.scrollTop = 0; }
                </script>
            </body>
</html>
```

Plus qu'a tester le nom de domaine sur le réseau interne en web : webdmz.mondomaine.com ou l'adresse de notre serveur web 192.168.1.233

Il nous reste plus qu'a télécharger des fichiers sous Amazon S3 en passant par notre serveur web :

Après avoir crée un compte Amazon AWS,

Dans le AWS Management Console

Rendez-vous dans S3

On crée un compartiment / bucket

puis on définie les propriétés, zone, accès

On peut désormais uploader des fichiers sur notre serveur cloud.

Afin de garantir un accès HTTP visible publiquement, on rend public avec les droits de lecture. On peut rendre l entièreté du bucket public ou simplement les fichiers que l'on souhaite.

VII - Configuration du Routeur Firewall Iptables

Dernière étape, et pas des moindres, la configuration du Routeur Firewall Iptables. Tout d'abord il nous faut configurer deux interfaces Ethernet sur ce routeur, une pointant vers le réseau DMZ et l'autre vers notre Réseau LAN. Cette devra avoir de configurer toutes les interfaces virtuel de chaque VLAN avec la même configuration que le serveur interne afin d'ajouter la norme dot1q et les packets VLAN.

En revanche pour l'interface pointant vers le réseau DMZ, il faudra définir une adresse ip fixe non virtuel sur le même réseau que notre DMZ, ici cela sera 192.168.1.234/29

Fichier de configuration interfaces routeurs /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

auto enp0s3.10
iface enp0s3.10 inet static
address 192.168.0.254
netmask 255.255.255.0
#network 192.168.0.0
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8

auto enp0s3.20
iface enp0s3.20 inet static
address 192.168.1.126
network 192.168.1.0
netmask 255.255.255.128
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8

auto enp0s3.30
iface enp0s3.30 inet static
address 192.168.1.158
network 192.168.1.128
netmask 255.255.255.224
vlan-raw-device enp0s3
```

```
dns-nameservers 8.8.8.8

auto enp0s3.40
iface enp0s3.40 inet static
address 192.168.1.190
network 192.168.1.160
netmask 255.255.255.224
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8

auto enp0s3.50
iface enp0s3.50 inet static
address 192.168.1.206
network 192.168.1.192
netmask 255.255.255.240
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8

auto enp0s3.60
iface enp0s3.60 inet static
address 192.168.1.222
netmask 255.255.255.240
network 192.168.1.208
vlan-raw-device enp0s3
dns-nameservers 8.8.8.8

auto enp0s3.70
iface enp0s3.70 inet static
address 192.168.1.230
netmask 255.255.255.248
network 192.168.1.224
```

vlan-raw-device enp0s3

dns-nameservers 8.8.8.8

auto enp0s8

iface enp0s8 inet static

address 192.168.1.234

netmask 255.255.255.248

network 192.168.1.232

gateway 192.168.1.238

dns-nameservers 8.8.8.8

post-up /etc/network/parefeu.sh

Nous remarquerons le « post-up /etc/network/parefeu.sh. » Cette commande permet de lancer le script iptables lorsque l'interface est fonctionnel.

Voici la configuration de parefeu Iptables. Afin de respecter au mieux les règles du parefeu, nous avons bloqué toutes les conexions entrantes, sortantes, traversantes pour ensuite définir uniquement ce que l'on souhaite autoriser.

Script parefeu.sh

Reset Iptables règles

iptables -F

iptables -X

iptables -t nat -F

iptables -t nat -X

Bloquer toute les connexions entrantes, sortantes, redirections

iptables -P INPUT DROP

iptables -P OUTPUT DROP

```
iptables -P FORWARD DROP

# Autoriser connexions établies

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Autoriser SSH  Depuis Reseau Interne

iptables -A INPUT -p tcp -i enp0s3 --dport 22 -j ACCEPT

# Autorisation DNS Depuis Reseau Interne

iptables -A FORWARD -p tcp -s 192.168.1.1 --dport 53 -j ACCEPT
iptables -A FORWARD -p udp -s 192.168.1.1 --dport 53 -j ACCEPT

# Ajout MASQUERADE

iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE

# Autorisation accès Internet depuis Proxy

iptables -A FORWARD -s 192.168.1.1 -j ACCEPT
iptables -A FORWARD -p tcp -s 192.168.1.1 --dport 80 -j ACCEPT
iptables -A FORWARD -p tcp -s 192.168.1.1 --dport 443 -j ACCEPT

# Autorisation ping entrant, sortant, mais non traversant pour plus de sécurité
iptables -A OUTPUT -p icmp -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT
#iptables -A FORWARD -p icmp -j ACCEPT
```

Autorisation accès DMZ depuis réseau interne

iptables -A FORWARD -s 192.168.1.0/25 -d 192.168.1.232/29 -j ACCEPT

iptables -A FORWARD -s 192.168.1.128/27 -d 192.168.1.232/29 -j ACCEPT

iptables -A FORWARD -s 192.168.1.160/27 -d 192.168.1.232/29 -j ACCEPT

iptables -A FORWARD -s 192.168.1.192/28 -d 192.168.1.232/29 -j ACCEPT

iptables -A FORWARD -s 192.168.1.208/28 -d 192.168.1.232/29 -j ACCEPT

iptables -A FORWARD -s 192.168.1.224/29 -d 192.168.1.232/29 -j ACCEPT

Autoriser le serveur proxy aller sur internet

iptables -A FORWARD -s 192.168.1.1 -j ACCEPT

VIII -Test complet du réseau

Démonstration DHCP sur différents VLAN

Démonstration aucun ping entre VLAN

Démonstration accès au Serveur Web depuis réseau Interne

Démonstration Téléchargement du ficher dans le bucket S3 depuis Serveur WEB

Démonstration NTP depuis réseau interne

Démonstration DNS & Proxy avec trames wireshark + Explication règles Firewall