SAE4.01

Livrable 5 - rapport final - version finale

Par

Kersual Arnaud, Ben-Hamida Naël, Laiadi Gibril, Estienne Quentin

Table des matières

1 Cadrage

1.1 Introduction et mise en contexte

Ce projet a pour but la conception et la mise en œuvre d'une infrastructure réseau documentée permettant d'accueillir les postes de travail "utilisateur" et "administrateur" d'une banque avec un certain nombre de services sécurisés à l'aide d'outils open-source.

Nous avions décidé de commencer par l'infrastructure améliorée pour par la suite arriver à une réalisation stable et exhaustive.Au final, nous avons terminé l'infrastructure avancée et avons implémenté une partie des services de l'infrastructure avancée (DoT,DoH, serveur de supervision du réseau sous Nagios Core,Annuraire LDAP).

1.2 Rappel Architecture envisagée et plan d'adressage

L'architecture a été adaptée et modifiée pour ne comporter qu'un seul routeur pour le niveau amélioré, nous facilitant ainsi l'implémentation de l'architecture. Les VLANs présents précédemment ont été changés pour des sous réseaux, sachant qu'il n'est pas possible d'émuler des VLANs au sein de libvirt (ou très compliqué). Dans la version finale nous avons rajouté de nouvelles machines telles qu'un serveur LDAP, un serveur NFS,un serveur de supervision du réseau utilisant NagiosCore ou encore un serveur de log.

cf l'annexe 1 : Schéma de l'architecture réseau et plan d'adressage au début du projet / final.

1.3 Récapitulatif des ressources matérielles utilisées

Ce deuxième tableau regroupe les serveurs qui implémentent les services suivant : DNS, DHCP, Nagios, NFS, serveur de logs, serveur LDAP, Base de données MariaDB.

Critères	VM-centrale	VM-client/Admin/Web externe/routeur	VM-Web interne
Stockage(en Go)	200	10	10
Ram (en Go)	25 Go	2	4
Nb coeurs (Processeur)	1	1	1
nombre de VMs	1	3	1

TABLE 1 – tableau récapitulatif des ressources matérielles utilisées par chaque machine virtuelle

Critères	Autres VM
Stockage(en Go)	10
Ram (en Go)	1
Nb coeurs (Processeur)	1
nombre de VMs	7

2 Documentation technique

La documentation technique complète est présente sur le wiki de notre projet à l'url : https ://109.31.139.7 . Nous donnons ici un résumé de la documentation technique de certains services.

2.1 Réseaux virtuels

Notre réseau est construit avec l'aide du logiciel virt-manager. Dans ce logiciel, nous créons et configurons nos machines virtuelles comportant des services ou servant de postes utilisateurs, en créant également nos interfaces réseau et en établissant les liens entre les machines concernées et leurs réseaux respectifs. Pour cela, nous devions créer des cartes réseau virtuelles, puis les configurer dans la machine virtuelle avec l'interface réseau correspondant à la carte créée.

2.2 Routeur

Étapes d'installation : Pour configurer une machine servant de routeur, nous devrons suivre le même processus que pour la création d'une machine virtuelle sur libvirt (voir les

étapes d'installation d'un réseau virtuel).

Après cette installation vous disposez désormais d'une machine virtuelle prête à accueillir un système d'exploitation. Procédez ensuite à l'installation du système d'exploitation en suivant attentivement chaque étape. En l'absence d'instructions spécifiques, optez systématiquement pour les choix par défaut proposés.

```
Language : English
Location : other/Europe/France
Locales : United States, en_US.UTF-8
Keyboard : French
Root Password : un mot de passe simple est conseillé, par exemple "root".
vous pouvez rajouter une clé ssh plus tard au besoin
User Account - Full Name : votre nom complet, par exemple "Jean Toto".
User Name : saisir votre nom de login UGA
User Password : saisir un mot de passe simple, par exemple "toto".
Vous pouvez redéfinir votre mot de passe plus tard
Partition disks : Guided - use entire disk
Partition disks : All files in one partition
Partition disks : Yes
Software Selection : vérifier que "Debian desktop" et "GNOME" ne sont pas cochés
et que "ssh server" est coché
Install GRUB : Yes
Device for boot loader : /dev/sda ou /dev/vda
(selon le choix du type d'OS que vous avez fait précédemment).
```

étapes de configuration : Une fois l'OS installé, accéder à votre VM routeur soit par le biais de virt-manager soient par connexion SSH.

Pour configurer votre routeur, il est d'abord nécessaire de déterminer le nombre de réseaux que nous prévoyons de créer et de connecter à notre routeur. Une fois l'étude des réseaux effectuée, nous devons les créer. Pour cela, ouvrez virt-manager, cliquez sur votre réseau actuel,
puis sur les onglets en haut, accédez à "Édition" -> "Détails de la connexion".

Ensuite, cliquez sur le bouton "+" en bas à gauche pour créer un nouveau réseau. Renommez ce réseau et sélectionnez le mode "isolated" pour maintenir l'isolement entre les réseaux (le routeur permettra la liaison entre eux).
Pour la configuration de l'adresse IP, si vous n'avez pas encore configuré de serveur DHCP, vous pouvez définir la plage d'adresses IP et libvirt se chargera d'attribuer des adresses IP dans cette plage pour ce réseau. De même, pour la configuration du DNS, si vous n'avez pas encore configuré de serveur DNS, vous pouvez attribuer un nom personnalisé à votre réseau. Une fois toutes vos configurations réseau réalisées, vous devez ajouter vos réseaux au routeur (voir étape suivante).

Finalement, il va falloir ajouter une/des interface(s) sur le routeur : Pour réaliser cela, commencez par double-cliquer sur votre machine virtuelle routeur. Une fenêtre s'ouvrira alors avec l'interface de ligne de commande de votre machine virtuelle. En haut de cette fenêtre, vous verrez un "i" bleu. En bas de cette page, vous trouverez l'option "Add Hardware" ; cliquez dessus. Ensuite, sélectionnez "Réseau" dans le menu déroulant qui apparaît. Vous y trouverez les réseaux que vous avez créés à l'étape précédente. Dans "Network Source", choisissez le réseau que vous souhaitez associer au routeur, puis cliquez sur "Terminer" et "Appliquer". Vous venez de créer une interface pour le réseau choisi avec une adresse IP attribuée dans l'une des plages que vous avez définies précédemment sur votre routeur. C'est par cette interface que vous pourrez communiquer avec les autres sous-réseaux.

2.3 Serveur DNS

étapes d'installation : Le serveur DNS peut être implémenté via le package bind9 trouvable via le gestionnaire de paquets "apt".
exécutez les commandes dans l'ordre :

```
apt update
apt upgrade
```

```
apt install bind9
```

étapes de configuration : Une fois installé, il va falloir configurer le DNS. Tout d'abord, nous devons configurer les différentes zones du réseau en nom de domaine.
Le fichier se trouve dans /etc/bind/named.conf.local exemple :

```
// La zone banque.fr
zone "banque.fr" {
    type master;
    file "/etc/bind/db.banque.fr"; // Fichier de résolution pour la zone Admin
    allow-transfer { any; }; // Permettre les transferts de zone
};

etc...
```

Par la suite pour chaque zone ajoutée, nous devrons créer un fichier db.[NOM-ZONE] qui permettra de faire l'affectation de <nom, adresse IP>
exemple le fichier db.banque.fr :

```
; BIND data file for admin zone
; Durée de vie par défaut
$TTL    604800
;                       Start of authority    Email
@       IN      SOA     banque.admin.  root.admin. (
                              3         ; Serial
                           3600         ; Refresh (1h)
                            600         ; Retry (10m,)
                          86400         ; Expire (24h)
                          600 )         ; Negative Cache TTL (10mn)
```

```
; nom complet du DNS pour le domande admin
@       IN      NS      banque.fr.  ; Utilisation du nom d'hôte

; Entrées pour la resolution <nom, adresse IP>
post1-admin           IN      A       [IP_STATION_ADMIN1]; Post de travail 1 de la zone
```

Il ne reste plus qu'à configurer sur chaque machine l'adresse IP du serveur DNS à qui ils vont envoyer leurs requêtes de résolution. Le fichier se trouve dans /etc/resolv.conf exemple :

```
namerserver [IP_SERVEUR_DNS] # utiliser nslookup ou dig pour tester
```

2.4 Serveur LDAP

Pour mettre en place le LDAP, commencez par installer le package suivant :

```
$ sudo apt install slapd ldap-utils
```

Une fois installé, vous devrez le configurer :

```
$ sudo dpkg-reconfigure slapd
```

Indiquez ensuite :

```
- pour nom DNS : banque.fr ;
- pour nom d'organisation : banque ;
- le mot de passe administrateur;
- choisissez le format de base par défaut : mdb ;
- No pour savoir si la base doit être supprimée quand slapd est purgé ;
- Yes pour déplacer l'ancienne base de données.
```

Créer un fichier ajout-groupEtUsers.ldif au possible dans un répertoire sur votre homedir

Ecrire l'ajout du domaine de base :

```
# Création du domaine de base
dn: dc=banque,dc=fr
changetype: add  # Ajout car il vient d'être créé
objectClass: top  # signifie au dessus dans la hiérarchie
objectClass: dcObject
objectClass: organization
o: banque
dc: banque
```

Ecrire l'ajout d'un groupe utilisateur "Users" :

```
# Création d'un groupe Users
dn: ou=Users,dc=banque,dc=fr
changetype: add
objectClass: organizationalUnit # Caractérise un groupe
ou: Users
```

Et pour ajouter un utilisateurs au sein de l'annuaire LDAP :

```
# Création d'une entrée utilisateur pour utilisateur1
dn: uid=utilisateur1,ou=Users,dc=banque,dc=fr
changetype: add
objectClass: inetOrgPerson
description: un utilisateur, le premier
cn: utilisateur1
sn: none
uid: user1
userPassword: [METTRE_MDP]
mail: user1@example.com
```

Pour finir, l'ajout d'un serviceAccount est nécessaire. Il permet à un service ou une application de s'authentifier auprès de l'annuaire LDAP et d'effectuer des actions au nom du service ou de l'application.

```
# Création d'une entrée,
# crée un nouveau compte de service avec le nom commun (cn) « serviceaccount »
dn: cn=serviceaccount,ou=Users,dc=banque,dc=fr
changetype: add
objectClass: inetOrgPerson
cn: serviceaccount
sn: serviceaccount
uid: serviceaccount
# et définit l'attribut userPassword sur « [MDP_SERVICE_ACCOUNT] »
userPassword: [MDP_SERVICE_ACCOUNT]
```

Pour lancer ce script "ajout-groupEtUsers.ldif" executer la commande suivante :

```
# ecrire votre mdp LDAP root à l'éxécution de la commande
$ sudo ldapmodify -x -D "cn=admin,dc=banque,dc=fr" -W -H ldap:// -f
add_content2.ldif
```

Nous disposons finalement d'un annuaire LDAP avec un groupe d'utilisateurs "User" et un utilisateur "user1" disposant d'un serviceAccount pour tout logiciel nécessitant l'utilisation du service d'authentification LDAP.

2.5 Serveur DHCP

Pour mettre en place le DHCP il faut tout d'abord installer et configurer le DHCP de Kea puis installer un relai DHCP sur le routeur

apt install curl apt-transport-https -y

installons le dépot de la version la plus récente de Kea puis installons le package.

```
curl -1sLf \  'https://dl.cloudsmith.io/public/isc/kea-2-4/setup.deb.sh' \
      | sudo -E bash

apt install isc-kea-dhcp4-server -y
```

changez la configuration du fichier /etc/kea/kea-dhcp4.conf en remplaceant par les lignes en dessous.

```
{
"Dhcp4": {
"interfaces-config": {
"interfaces": ["INTERFACE"]
},

"lease-database": {
"type": "memfile",
"persist": true,
"name": "/var/lib/kea/kea-leases4.csv",
"lfc-interval": 3600
},

"renew-timer": 15840,
"rebind-timer": 27720,
"valid-lifetime": 31680,

"option-data": [
{
"name": "domain-name-servers",
"data": "IP_DNS"
},

{
"name": "domain-search",
"data": "DOMAINE_DNS"
}
],

"subnet4": [
// reseau utilisateurs
```

```
{
"subnet": "IP_CIDR_RESEAU_UTILISATEUR",
"pools": [ { "pool": "X.X.X.X - X.X.X.X" } ],
"option-data": [
{
"name": "routers",
"data": "X.X.X.X"
}
]

// Add reservations here
},
// reseau admin
{
"subnet": "IP_CIDR_RESEAU_ADMINISTRATEUR",
"pools": [ { "pool": "X.X.X.X - X.X.X.X" } ],
"option-data": [
{
"name": "routers",
"data": "X.X.X.X"
}
]

// Add reservations here
},
// reseau du serveur dhcp ( conseille dans les bonnes pratiques , pour que
    le
serveur "comprenne la topologie du reseau")
{
"subnet": "IP_CIDR_RESEAU_DHCP",
"pools": [ { "pool": "X.X.X.X - X.X.X.X" } ],
"option-data": [
{
"name": "routers",
"data": "X.X.X.X"
}
]

// Ajoutez des reservations d'adresses IP ici.
```

```
}

// Ajoutez des sous reseaux ici
]
}
}
```

avec : - INTERFACE : l'interface sur laquelle doit fonctionner le DHCP. Dans notre cas, on ne rentre qu'une interface mais si l'architecture change et que le serveur DHCP doit répondre sur plusieurs interfaces, ajoutez simplement un espace puis rentrez la ou les autres interfaces. - IP DNS / DOMAINE DNS : l'adresse IP et le nom de domaine du serveur DNS que vous voulez utiliser pour associer les IP que donnent le DHCP à un nom. Si vous n'avez pas encore déployé de serveur DNS n'écrivez pas les lignes suivantes : - IP CIDR RESEAU UTILISATEUR/ADMINISTRATEUR/DHCP : l'adresse du réseau utilisateur, administrateur et DHCP en notation CIDR (ex : 192.168.12.3/25)

- X.X.X.X :

- “"pools" : ["pool" : "X.X.X.X - X.X.X.X"]“ :

l'intervalle d'adresses IP qui peut être allouée par le DHCP.

- “"name" : "routers","data" : "X.X.X.X"“ :

l'adresse IP du routeur de l'interface qui est sur le réseau de la machine qui demande une adresse.

Il faut ensuite installer le service de relais DHCP sur le routeur

```
apt install isc-dhcp-relay
```

Puis changez la configuration du fichier /etc/default/isc-dhcp-relay

```
# Defaults for isc-dhcp-relay initscript
# sourced by /etc/init.d/isc-dhcp-relay
# installed at /etc/default/isc-dhcp-relay by the maintainer scripts

#
# This is a POSIX shell fragment
#

# What servers should the DHCP relay forward requests to?
# Remplacez X.X.X.X par l'adresse de votre serveur DHCP
SERVERS="X.X.X.X"

# On what interfaces should the DHCP relay (dhrelay) serve DHCP requests?
INTERFACES=""

# Additional options that are passed to the DHCP relay daemon?
# -id renseigne les interfaces sur lesquelles le relais doit ecouter les
   requ tes DHCP
# -iu renseigne l'interface sur laquelle le relais doit ecouter les
   reponses DHCP (interface du r seau DHCP)
#remplacez enpXs0 par le nom des interfaces concern es
OPTIONS="-id enpXs0 -id enpXs0 -iu enpXs0"
```

Vous devriez maintenant pouvoir obtenir une adresse ip depuis une machine sur un autre sous réseau que le DHCP.

3 Tests sur l'infrastructure

De manière générale, les tests sur l'infrastructure sont détaillés dans le livrable 3, nous montrons ici un résumé.

3.1 Accès au tableau de bord Nagios

Nous avons mis en place un tableau de bord Nagios nous permettant d'analyser l'état des machines du réseau beaucoup plus facilement. Dans cet exemple, le serveur effectue différentes vérifications sur lui même et sur la machine virtuelle administrateur.

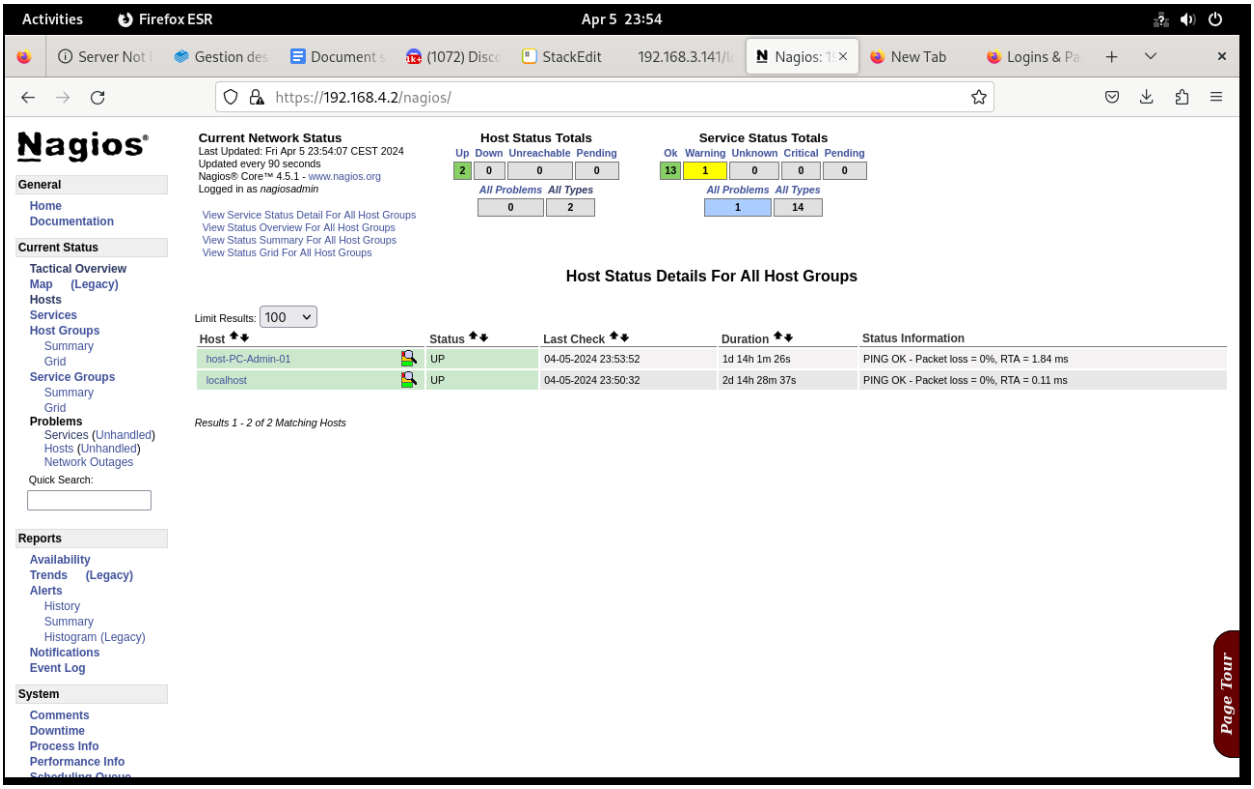


FIGURE 1 – capture d'écran du tableau de bord

3.2 test de ping entre machines

```
Connexion SSH au post1-client ...
Connecté à la zone post1-client !
|-------------------------|
|Les tests vont commencer|
|-------------------------|
------------------------------------------------------------------
|---------|
|   OK    |  post1-client(192.168.1.2) --> serveur-web-interne(Address: 192.168.3.141)
|---------|
------------------------------------------------------------------
|---------|
|   OK    |  post1-client(192.168.1.2) --> serveur-dns-interne(Address: 192.168.1.141)
|---------|
------------------------------------------------------------------
|---------|
|   OK    |  post1-client(192.168.1.2) --> post1-admin(Address: 192.168.2.2)
|---------|
------------------------------------------------------------------
|---------|
|   OK    |  post1-client(192.168.1.2) --> serveur-dhcp(Address: 192.168.2.130)
|---------|
------------------------------------------------------------------
|---------|
|   OK    |  post1-client(192.168.1.2) --> routeur-central(Address: 192.168.1.142)
|---------|
------------------------------------------------------------------
|---------|
|   OK    |  post1-client(192.168.1.2) --> serveur-nfs(Address: 192.168.3.136)
|---------|


Connexion SSH au serveur-dhcp ...
Connecté à la zone serveur-dhcp !
|-------------------------|
|Les tests vont commencer|
|-------------------------|
------------------------------------------------------------------
|---------|  DNS resolution error !
|  ERROR  |  serveur-dhcp(192.168.2.130) --> serveur-web-interne()
|---------|
------------------------------------------------------------------
|---------|  DNS resolution error !
|  ERROR  |  serveur-dhcp(192.168.2.130) --> serveur-dns-interne()
|---------|
------------------------------------------------------------------
|---------|  DNS resolution error !
|  ERROR  |  serveur-dhcp(192.168.2.130) --> post1-admin()
|---------|
------------------------------------------------------------------
```

FIGURE 2 – capture d'écran des tests de Ping

3.3 tests de résolution DNS

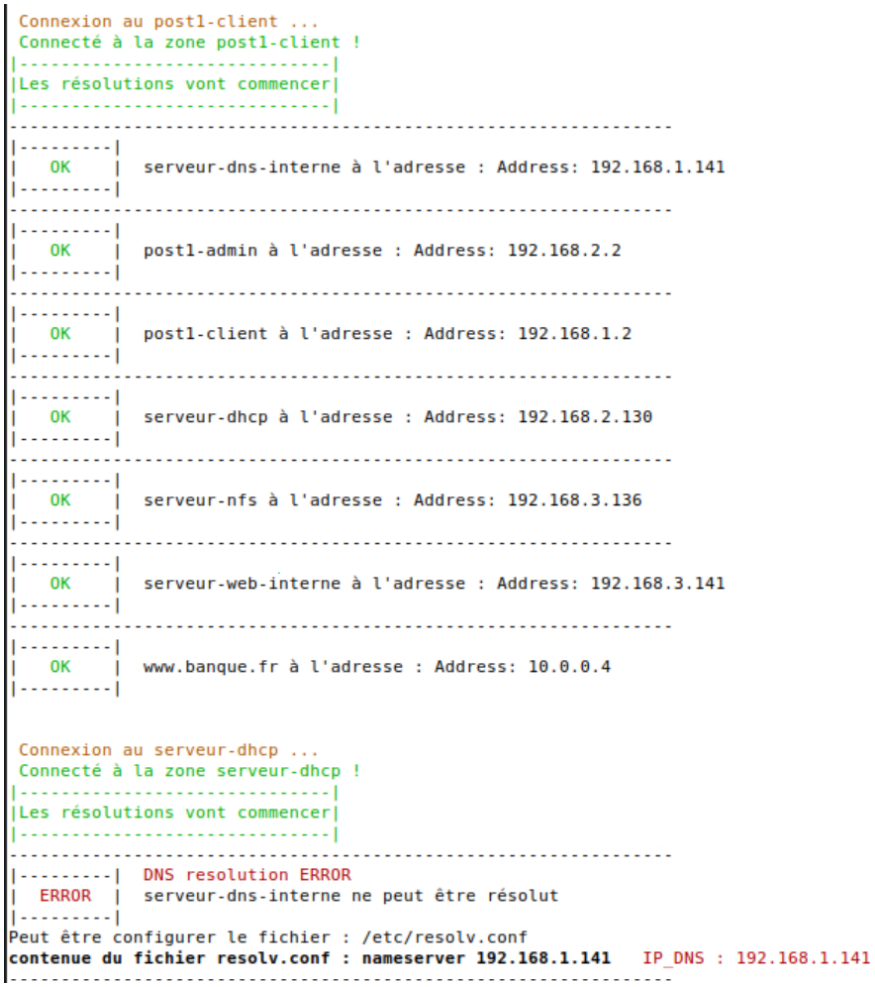


FIGURE 3 – capture d'écran des tests de résolution DNS

4 Aspects liés à la sécurité

Pour se défendre des tentatives d'intrusions, nous avons mis en place un pare-feu situé sur notre routeur. Nous avons autorisé uniquement les ports utiles et utilisés par les différents services, c'est une application des moindres privilèges, également appliquée sur les stations. Par exemple, les postes administrateurs sont censés avoir accès à toutes les autres machines, pour permettre cela, nous autorisons toutes les connexions provenant de ces postes. Au contraire, aucune des autres machines de notre infrastructure réseau n'est censée avoir accès aux machines administrateurs. Nous avons également prévu des règles pour empêcher le flood udp, icmp...

cf l'annexe 1 pour voir les règles de pare feu en détail.

Nous avons téléchargé la majorité de nos logiciels utilisés avec le gestionnaire de fichiers de Debian. Nous n'avons donc pas à gérer la majorité des mises à jour à la main et pouvons nous reposer sur les mises-à-jour de Debian. Debian étant la distribution Linux que nous avons choisie pour sa sécurité et sa réactivité à corriger les failles de sécurité.

Non n'avons pas autorisé les connexions SSH sur les machines directement en root par mot de passe mais uniquement par clé ssh. Le mot de passe pour accéder au Root est également long (20 caractères) et composé de différents chiffres, caractères et signes.

Le serveur de logs, au cœur de toute infrastructure de sécurité, joue un rôle pivot en assurant la centralisation, l'analyse et la gestion des journaux générés par les divers composants du réseau. En tant que point central de collecte des données de journalisation, il offre une vue consolidée des activités réseau, ce qui facilite la détection précoce des menaces et des anomalies.

Exemple de différents logs centralisés :

```
root@serveur-log:~# ls /var/log/remote
VM-Admin-1  VM-Admin-1.log  VM-Client-1.log  debian  debian.log  post1-
   admin.log  serveur-dns-interne.log
```

En cas d'incident de sécurité, le serveur de logs devient une ressource précieuse pour mener des investigations forensiques approfondies, permettant ainsi une compréhension approfondie de l'incident et la mise en œuvre de mesures correctives pour renforcer la posture de sécurité globale du réseau. En cas d'attaque, un attaquant peut effacer les traces en supprimant les journaux de la machine attaquée. Cependant, notre serveur de journaux permet d'avoir une copie intacte des journaux, préservant ainsi l'intégrité des données en cas de besoin.

Enfin notre serveur NFS nous permet de réduire significativement l'utilisation de clés USB et de disques externes, qui sont plus difficiles à tracer et présentent des risques accrus en termes de sécurité informatique. En effet, ces supports externes sont plus susceptibles de contenir des virus informatiques, ce qui peut compromettre la sécurité de nos données plus facilement. Il suffit donc de bien sécuriser un serveur de fichier unique et cela renforce la sécurité générale quant à la gestion de fichiers.

Exemple de dossiers obtenus par NFS :

```
> root@post1-client:~# ls /mnt/nfs/
users

> root@post1-admin:~# ls mnt/nfs/
documents
```

5 Conclusion

5.1 Retour d'expérience

Après avoir terminé ce projet, tout l'équipe s'accorde pour dire que celui ci nous a permis d'acquérir énormément de connaissances sur l'administration système en très peu de temps. Nous avons également beaucoup gagné en autonomie puisque nous devions guider nos travaux nous mêmes du début à la fin. Cette expérience n'aurait pas pu être obtenu aussi rapidement autrement que par la façon dont le projet s'est déroulé. Découvrir par soi-même le fonctionnement de nouveaux services,protocoles ou encore le fonctionnement de Linux était très gratifiant. Malgré cela, le fait que nous ne connaissions pas le fonctionnement complet de la quasi integralité des services à mettre en place nous a poussé à chaque fois à faire une sorte de "formation accélérée". Le fait d'apprendre de cette manière fait que nous n'avons pas integré de manière durable les connaissances acquises, car nous n'avions tout simplement pas le temps de le faire, dès qu'un nouveau service était fonctionnelle, il fallait passer au suivant. Heureusement, pour contrebalancer ce sentiment, écrire une doc de la mise en place d'une infrastructure nous a poussé à comprendre comment fonctionnait chaque service pour pouvoir l'expliquer le plus clairement possible. Nous avons donc acquéri de nombreuses connaissances que l'on pourra restituer ultérieurement.
Au final, les consignes à propos du contenu des livrables étaient parfois contracdictoires et assez vagues ce qui a causé une grande difficulté pour le groupe à rendre des livrables en cohérence avec les attentes des professeurs. Rendre compte de manière exhaustive de l'integralité du travail accompli à travers des livrables écrits nous semblait très compliqué et cela a pu créer de la frustration au sein du groupe.
En conclusion, cette SAE a été pour nous l'occasion d'énormément agrandir nos connaissances par nous même et de vraiment devenir autonome pour l'administration de systèmes d'information, ce qui est très gratifiant. D'un autre côte, la quantité de travail à fournir et

le fait que ce travail était parfois laborieux a pu amener beaucoup de frustration et de stress au sein du groupe. Nous avons pu réduire l'importance de ces problèmes en appliquant les méthodes de communciations assertives et les méthodes de gestion de conflit apprises en cours de communication.

5.2 Pistes d'amélioration

Étant limités en ressources, nos perspectives d'amélioration se concentrent sur les services déjà présents au sein de notre infrastructure réseau.

En premier lieu, nous envisageons d'améliorer le service LDAP pour prendre en charge les utilisateurs UNIX et ainsi permettre l'authentification sur le système. Nous envisageons également de sécuriser le service LDAP en mettant en place du LDAPS avec un certificat.

En deuxième lieu, nous envisageons l'installation et la configuration du service Kerberos pour assurer la sécurité de l'authentification des utilisateurs, même si le réseau n'est pas forcément sécurisé.

En troisième lieu, nous envisageons la mise en place d'un serveur CAS pour permettre une authentification unique sur tous les services de l'infrastructure utilisant le CAS.

En quatrième lieu, nous souhaitons ajouter un serveur SSH bastion, qui centralise le point d'entrée à un réseau privé via une connexion SSH, permettant ainsi de se connecter aux différentes instances et de surveiller de plus près les accès des utilisateurs à l'infrastructure.

Enfin, nous aimerions installer SSSD, une alternative à NSCD fournie par défaut, afin d'améliorer la sécurité de l'authentification et de l'autorisation.

1 Annexe

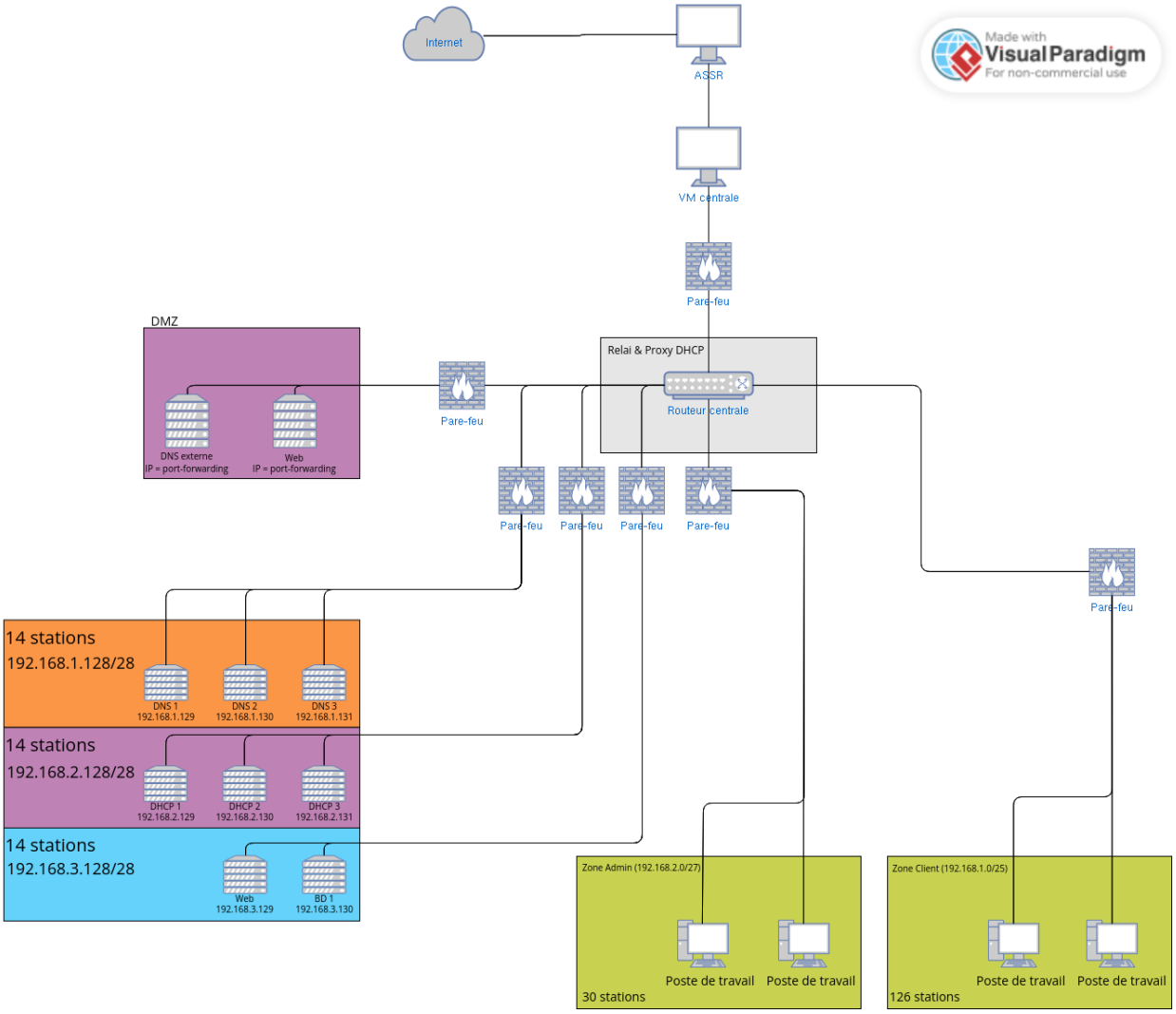


FIGURE 4 – Schéma de l'architecture réseau et plan d'adressage au début du projet

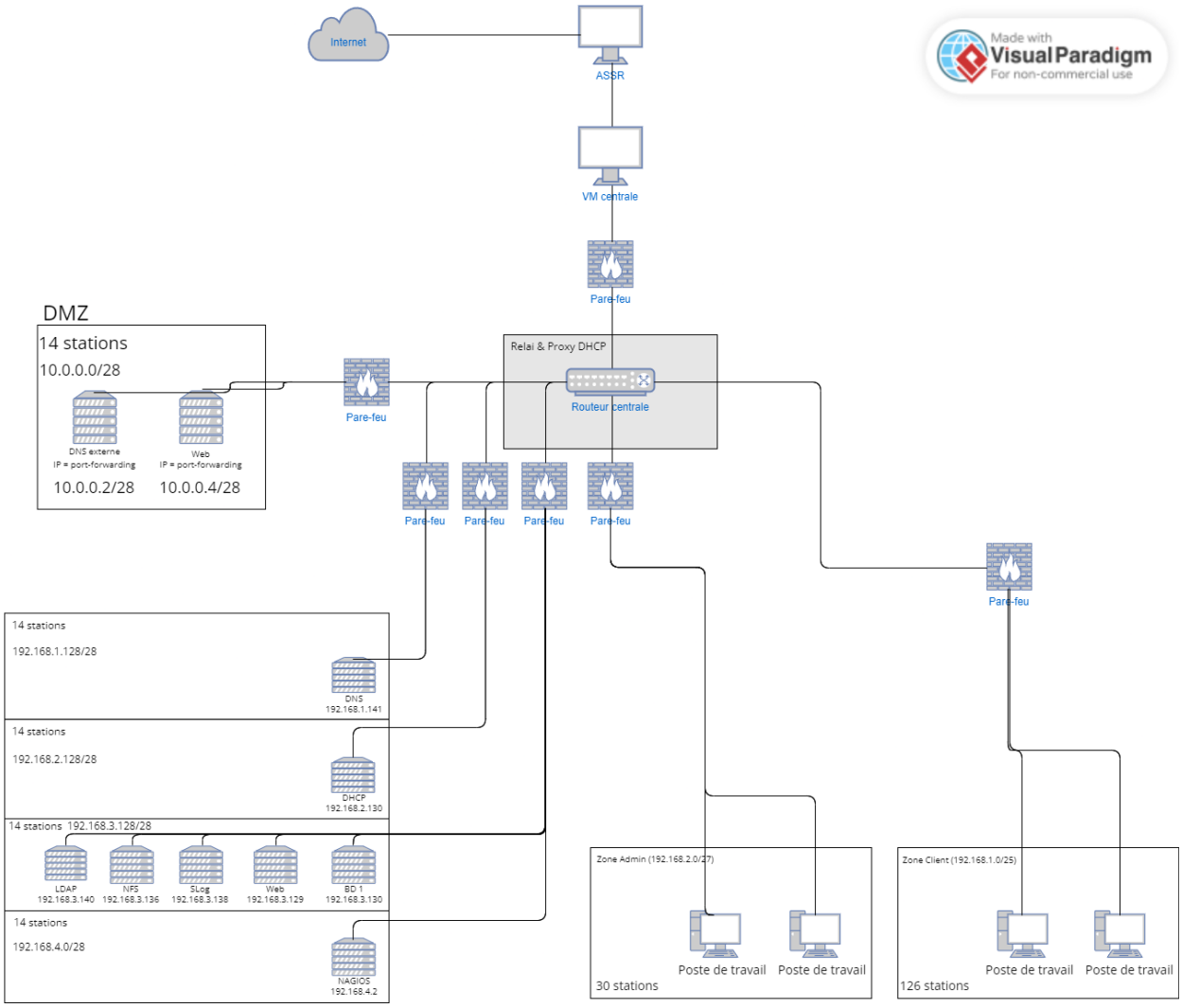


FIGURE 5 – Schéma de l'architecture réseau et plan d'adressage final

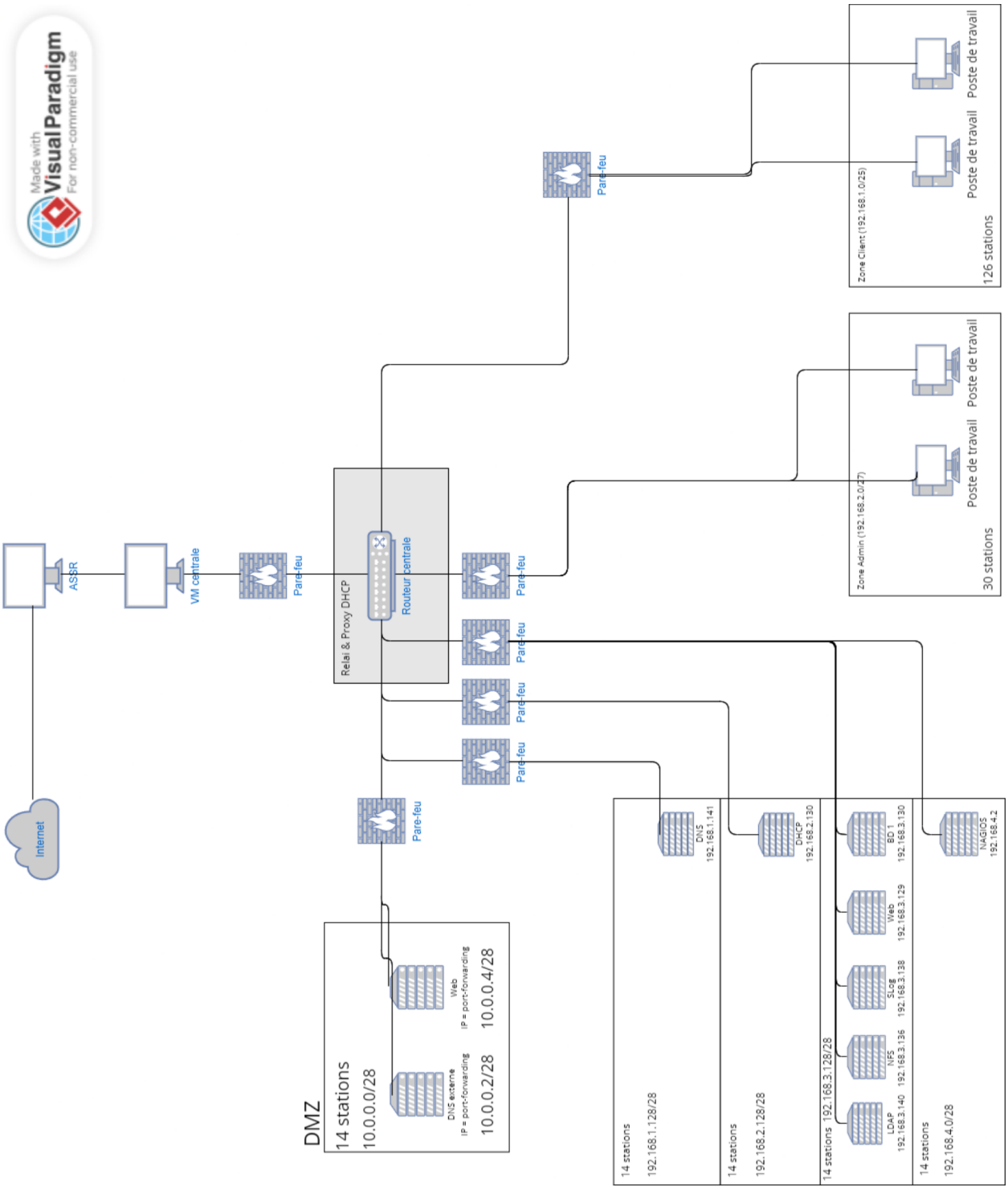
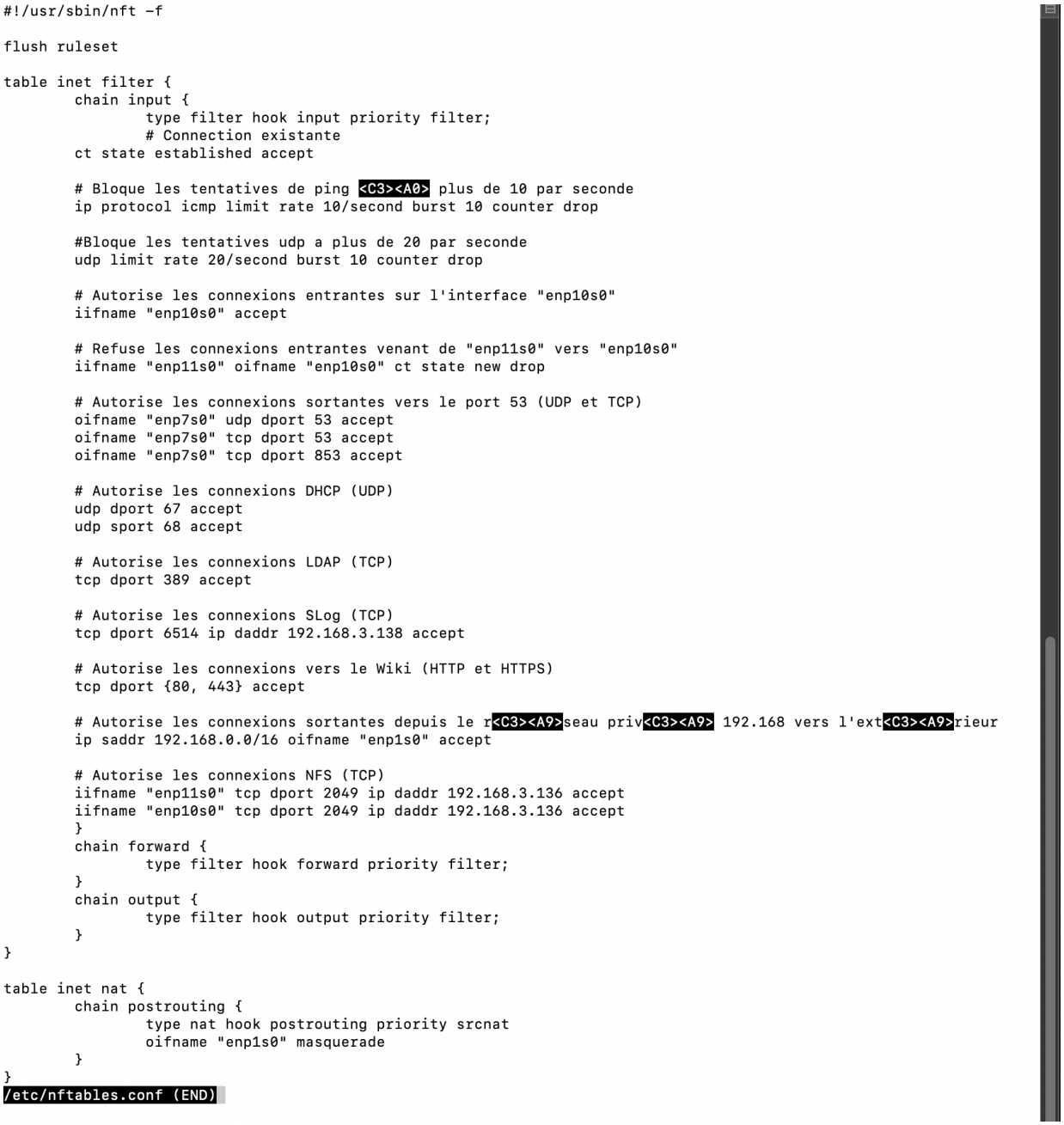


FIGURE 6 – Schéma de l'architecture réseau et plan d'adressage final (agrandi)



```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
        chain input {
                type filter hook input priority filter;
                # Connection existante
        ct state established accept

        # Bloque les tentatives de ping <C3><A0> plus de 10 par seconde
        ip protocol icmp limit rate 10/second burst 10 counter drop

        #Bloque les tentatives udp a plus de 20 par seconde
        udp limit rate 20/second burst 10 counter drop

        # Autorise les connexions entrantes sur l'interface "enp10s0"
        iifname "enp10s0" accept

        # Refuse les connexions entrantes venant de "enp11s0" vers "enp10s0"
        iifname "enp11s0" oifname "enp10s0" ct state new drop

        # Autorise les connexions sortantes vers le port 53 (UDP et TCP)
        oifname "enp7s0" udp dport 53 accept
        oifname "enp7s0" tcp dport 53 accept
        oifname "enp7s0" tcp dport 853 accept

        # Autorise les connexions DHCP (UDP)
        udp dport 67 accept
        udp sport 68 accept

        # Autorise les connexions LDAP (TCP)
        tcp dport 389 accept

        # Autorise les connexions SLog (TCP)
        tcp dport 6514 ip daddr 192.168.3.138 accept

        # Autorise les connexions vers le Wiki (HTTP et HTTPS)
        tcp dport {80, 443} accept

        # Autorise les connexions sortantes depuis le r<C3><A9>seau priv<C3><A9> 192.168 vers l'ext<C3><A9>rieur
        ip saddr 192.168.0.0/16 oifname "enp1s0" accept

        # Autorise les connexions NFS (TCP)
        iifname "enp11s0" tcp dport 2049 ip daddr 192.168.3.136 accept
        iifname "enp10s0" tcp dport 2049 ip daddr 192.168.3.136 accept
        }
        chain forward {
                type filter hook forward priority filter;
        }
        chain output {
                type filter hook output priority filter;
        }
}

table inet nat {
        chain postrouting {
                type nat hook postrouting priority srcnat
                oifname "enp1s0" masquerade
        }
}
/etc/nftables.conf (END)
```

FIGURE 7 – Capture d'écran des règles de pare feu appliquées sur le routeur