ADRAR FORMATION

Mis en place d’une DMZ

MARAVAL Liam
19/11/2024

Ce document sera décomposé en plusieurs chapitres :

1. Contexte : Définition des besoins ainsi que le choix d'une solution en réponse à la demande client.
2. Procédure technique : Procédure technique de mise en place de la solution
3. Mise en place d'une DMZ suivant les recommandations ANSSI
4. Conclusion
5. Annexes

Table des matières

1. Contexte 2
1.1 Demandes du client 2
1.2 Evolution de l'infrastructure 3
1.3 Choix des solutions 4
1.4 Sécurisation des accès 4
2. Procédure technique 6
2.1 Configuration du serveur Web 6
2.2 Configuration du serveur MariaDB 7
2.4 Configuration de la règle NAT 14
2.3 Configuration des règles de pare feu 15
3. Mise en place d'une DMZ externe avec Reverse Proxy 17
3.1 Evolution de l'infrastructure en suivant les recommandation ANSSI 17
3.2 Evolution des règles de pare feu 18
3.3 Mise en place de la solution 18
4. Conclusion 22
5. Annexes 23
5.1 Documentations et références 23

1. Contexte

1.1 Demandes du client

Nous sommes contactés par l'ADRAR afin de mettre en place une DMZ qui servira de zone tampon pour un serveur Web exposé sur internet.

Nous devons également configurer une base de données que le serveur Web sera en mesure d'interroger.

Voici les règles de sécurité à mettre en place :

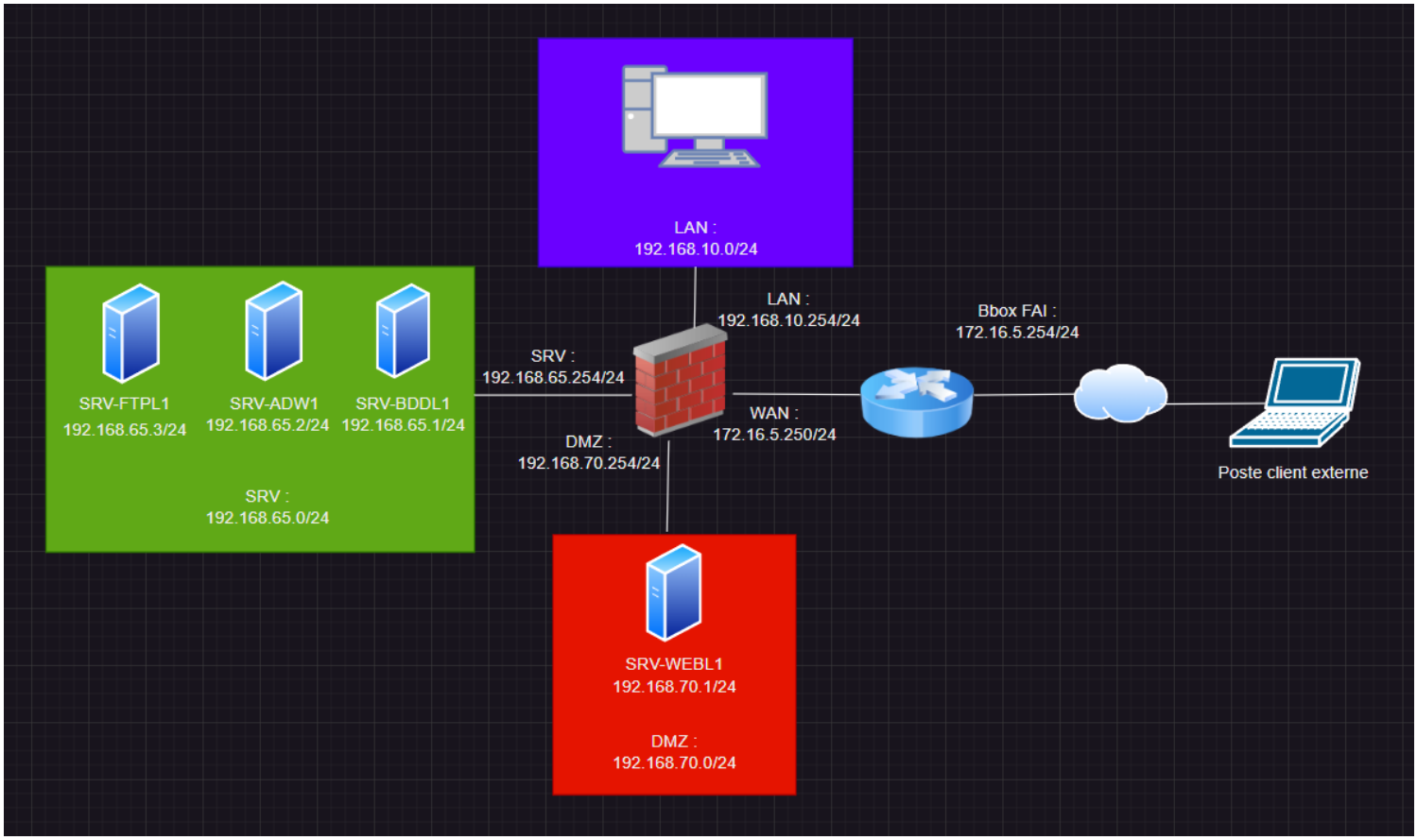
- ✓ Autoriser les accès au serveur Web depuis internet et depuis le LAN
- ✓ Autoriser les accès SFTP sur le serveur de la DMZ depuis le LAN
- ✓ Autoriser les requêtes au serveur MySQL depuis le serveur Web de la DMZ
- ✓ Autoriser les accès Internet depuis le LAN et la DMZ en passant par le pare-feu
- ✓ Interdire tout autre accès au LAN depuis internet ou la DMZ.

Voici l'infrastructure en place actuellement :

- Une sortie internet via une Bbox en 172.16.5.254/24
- Un serveur SFTP
- Un serveur Active Directory/DNS avec un domaine « **adrar.local** »
- Un réseau global pour les postes utilisateurs et les serveurs
- Un firewall Opnsense avec un service DHCP pour le LAN

1.2 Evolution de l'infrastructure

Voici le schéma de la nouvelle infrastructure :



Nous allons tout d'abord découper notre réseau global en 4 réseaux différents :

- Un réseau utilisateurs en 192.168.10.0/24
- Un réseau serveurs en 192.168.65.0/24
- Un WAN qui restera sur le réseau historique en 172.16.5.0/24
- Une DMZ pour le serveur Web exposée sur internet en 192.168.70.0/24

Ces réseaux seront rattachés aux interfaces de l'Opnsense en place.

Nous avons mis en place un réseau dédié pour les serveurs, cela étant plus sécurisé, évolutif et cohérent de séparer le réseau utilisateurs des serveurs.

Ensuite, nous allons ajouter les 2 serveurs demandés :

NOM	IP	Role	OS
SRV-WEBL1	192.168.70.1	Serveur web Apache2	Debian 12
SRV-BDDL1	192.168.65.1	Serveur de base de données MariaDB	Debian 12

Le serveur Web devait être initialement installé avec le serveur SFTP existant, mais afin de dissocier les usages, comme le recommande l'ANSSI, celui-ci seras configuré sur un serveur dédié.

Une règle NAT sera configurée sur l'Opnsense afin de rediriger le trafic HTTPS arrivant sur l'interface WAN vers notre serveur WEB.

IP Cible	Port Cible	IP de redirection	Port de redirection
IP interface WAN : 172.16.5.250/32	Port HTTPS : TCP 443	IP Serveur WEB : 192.168.70.1/32	Port HTTPS : TCP 443

Une fois cette règle NAT en place, Opnsense nous générera la règle de pare feu autorisant le trafic automatiquement.

Concernant les nouvelles règles de pare feu mises en place afin de sécuriser les nouveaux flux, veuillez-vous référer au chapitre **1.4 Sécurisation des accès**

1.3 Choix des solutions

Concernant la base de données, nous avons opté pour **« MariaDB »**.
Voici les raisons pour lesquelles nous l'avons choisi :

- Solution open source
- Grande communauté et documentation facile à trouver
- Evolution de MySQL
- Moderne et compatible avec les solutions Cloud
- Sécurité accrue
- Mise a jours régulière et support disponible

Pour le serveur Web nous avons opté pour **« Apache 2 ».**
Ce choix repose sur sa stabilité, sa flexibilité, ses mises à jour régulières ainsi que sa large documentation.

Enfin, les OS utilisés seront Debian 12.
Reconnu pour sa stabilité, sa simplicité d'administration, sa communauté très présente ainsi que ses mises à jour de sécurité régulières, celui-ci facilitera les futures évolutions et l'administration au quotidien.

1.4 Sécurisation des accès

Pour sécuriser notre infrastructure et notamment le serveur Web exposé sur internet nous allons ajouter plusieurs règles de pare feu.

Sur l'interface **WAN** nous autorisons les connexions HTTPS.
Celles-ci seront redirigées, via la règle NAT, sur notre serveur Web situé dans la DMZ.

Sur notre **DMZ,** nous allons autoriser la communication avec notre serveur MariaDB ainsi que les flux HTTPS vers internet comme demandé par l'ADRAR.

Sur l'interface **LAN** nous avons autorisé les connexions SSH/SFTP sur les serveurs de la **DMZ** et du réseau **SRV**. Le port par défaut du SFTP a été modifié pour des raisons de sécurité, celui est désormais sur le port **« 4422 ».**

Sur l'interface **SRV** nous avons autorisé les résolution DNS ainsi que le trafic HTTPS notamment pour la gestion des mises à jours des serveurs.

Voici un tableau récapitulatif des règles de pare feu :

LAN				
Protocol	IP Source	Port source	IP Destination	Port Destination
TCP	192.168.10.0/24	any	any	443
TCP	192.168.10.0/24	any	any	80
UDP	192.168.10.0/24	any	192.168.65.2/32	53
UDP	any	68	any	67
TCP	192.168.10.0/24	Any	192.168.65.0/24	22
TCP	192.168.10.0/24	Any	192.168.70.0/24	22
TCP	192.168.10.0/24	Any	192.168.65.3/32	4422

WAN				
Protocol	IP Source	Port source	IP Destination	Port Destination
TCP	any	any	192.168.70.1/32	443
UDP	172.16.5.0/24	Any	192.168.65.2/32	53

DMZ				
Protocol	IP Source	Port source	IP Destination	Port Destination
UDP	192.168.70.0/24	Any	192.168.65.2/32	53
TCP	192.168.70.0/24	any	any	443
TCP	192.168.70.0/24	Any	192.168.65.1/32	3306

SRV				
Protocol	IP Source	Port source	IP Destination	Port Destination
TCP	192.168.65.0/24	Any	192.168.65.2/32	53
TCP	192.168.65.0/24	Any	Any	443

2. Procédure technique

Afin de ne pas surcharger la documentation et d'éviter les doublons, nous ne reviendrons pas sur certaines procédures déjà établies pour l'ADRAR dans de précédents projets.

Voici les prérequis techniques nécessaires à la mise en place de notre solution :

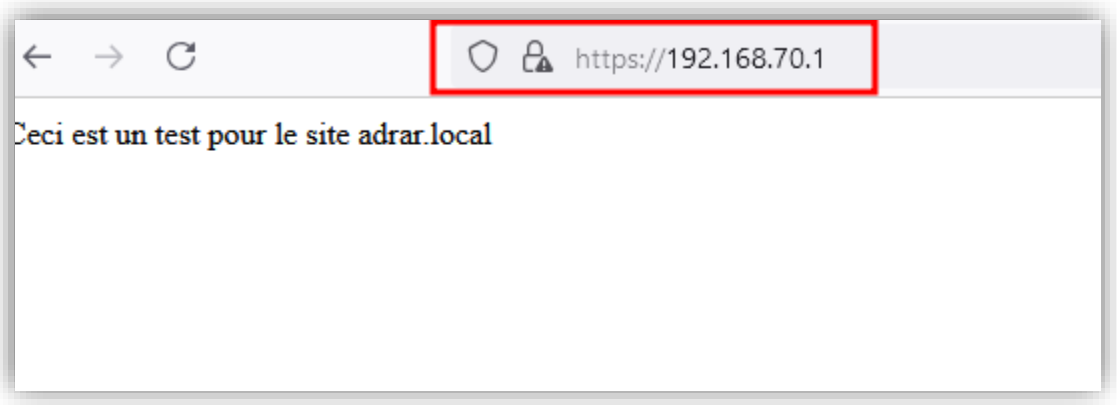
- ✓ Configurer et sécuriser un serveur Web pour le site « adrar.local »
- ✓ Créer des règles de pare feu Opnsense
- ✓ Générer des certificats via l'autorité de certification sur l'Opnsense

Pour plus de détails, merci de vous référer aux précédentes documentations rendues à l'ADRAR.

2.1 Configuration du serveur Web

Comme mentionné plus haut, la configuration de notre serveur Web Apache2 est en place et sécurisée via SSL.

Voici un aperçu de la page Web :



Nous allons installer le client MariaDB sur le serveur afin de pouvoir initier les connexions SQL, une fois que le serveur de base de données sera configuré.

```
apt install mariadb-client
```

Nous installons également PHP qui servira à générer du contenu sur le serveur Web :

```
apt install -y php8.3
```

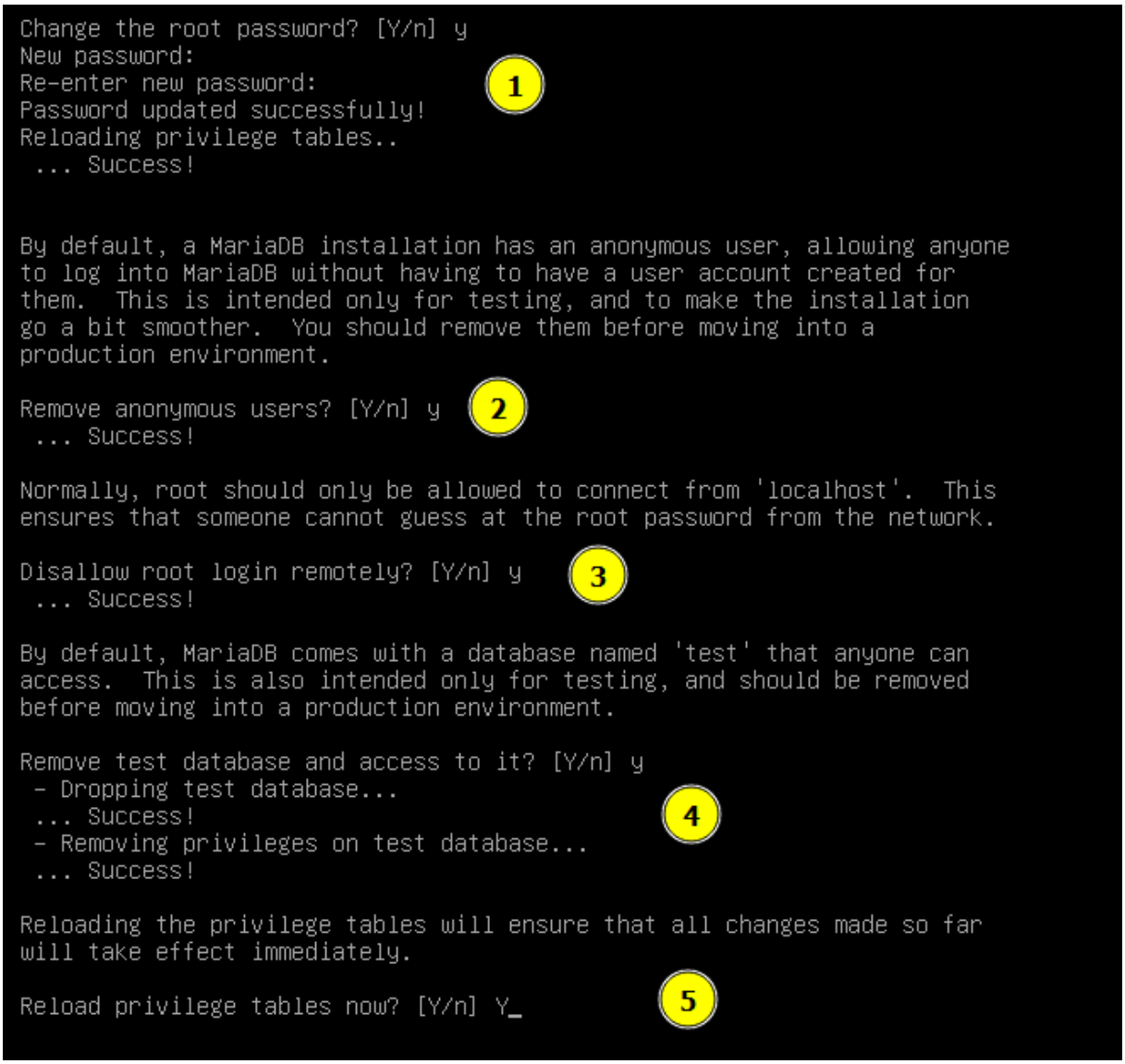
2.2 Configuration du serveur MariaDB

Par mesure de sécurité, les différents mots de passe n'apparaitront pas en clair dans cette procédure, ceux-ci étant renseignés dans le KeePass de l'entreprise.

Concernant l'installation de MariaDB merci de vous référer au lien : https://docs.vultr.com/how-to-install-mariadb-on-debian-12

Maintenant que l'installation est effectuée, nous allons lancer le script **« MariaDB secure installation ».**

Celui-ci va nous demander plusieurs options à choisir pour affiner la sécurité de notre base de donnée.



1. Changement du mot de passe **« root »**
2. Empêcher les connexions anonymes
3. Empêcher la connexion avec le compte **« root »**
4. Supprimer la base de test ainsi que ses accès
5. Mis à jour des droits

Nous créons ensuite une base de donnée nommée **« bd1 »**

```
CREATE DATABASE bd1;
```

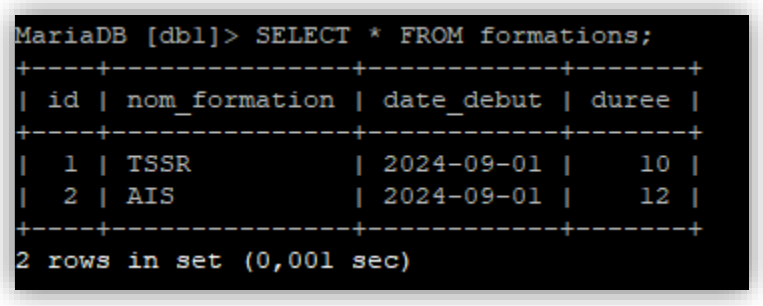
Nous allons maintenant créer une table **« formation »** qui contiendra des colonnes pour les noms, dates de début et durée de formation proposées par l'ADRAR :

```
CREATE TABLE formations (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom_formation VARCHAR(100),
    date_debut DATE,
    duree INT, -- Durée de la formation en mois
);
```

Une fois cela fait, nous allons maintenant la peupler :

```
INSERT INTO formations (nom_formation, date_debut, duree)
VALUES
('TSSR', '2024-09-01', 10),
('AIS', '2024-09-01', 12);
```

Voici le résultat :



```
MariaDB [dbl]> SELECT * FROM formations;
+----+---------------+------------+-------+
| id | nom_formation | date_debut | duree |
+----+---------------+------------+-------+
|  1 | TSSR          | 2024-09-01 |    10 |
|  2 | AIS           | 2024-09-01 |    12 |
+----+---------------+------------+-------+
2 rows in set (0,001 sec)
```

Nous allons maintenant créer un utilisateur **« admin »** pour se connecter aux bases de données :

```
CREATE USER admin@localhost INDENTIFIED BY '******'
```

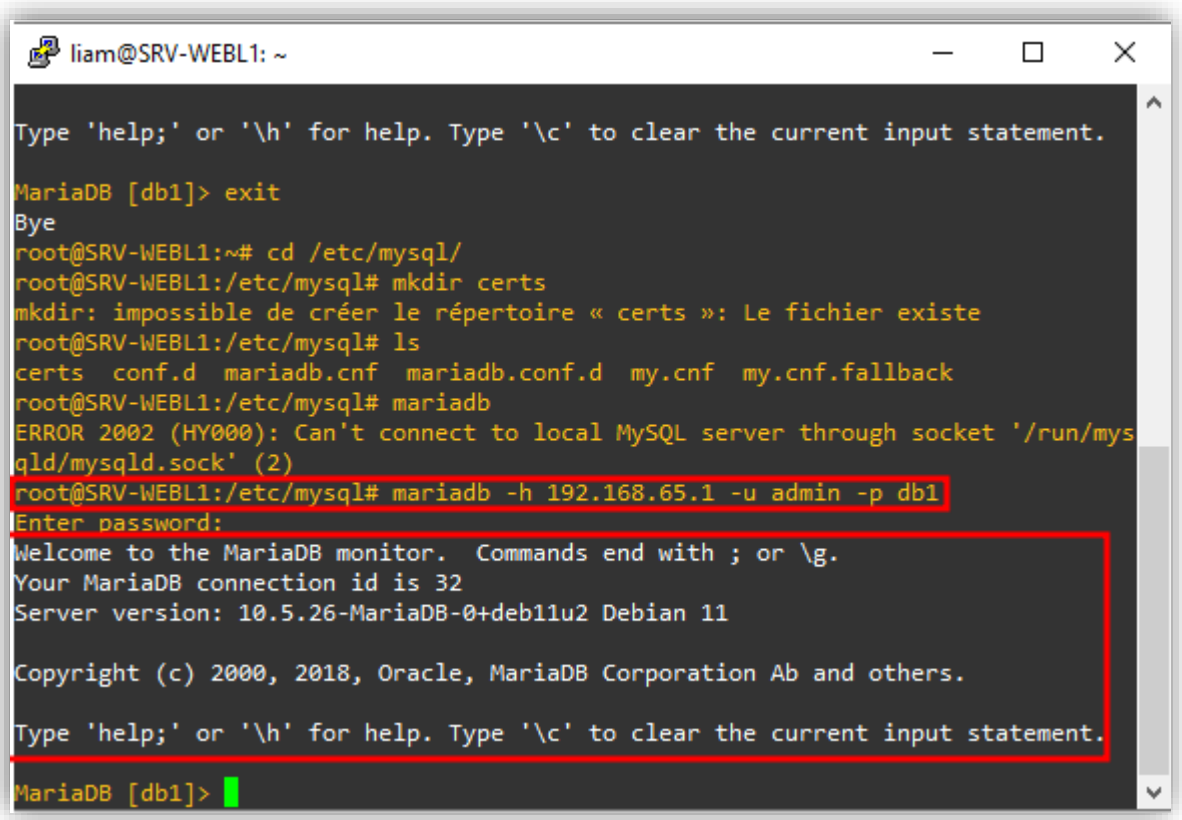
Nous lui assignons ensuite les droits sur les bases de données en indiquant les IP de connexion autorisée (celle du serveur Web dans notre cas) :

```
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'192.168.70.1' IDENTIFIED BY '******';
FLUSH PRIVILEGES;
```

Il faut aussi autoriser la connexion via les IP distantes dans **« /etc/mysql/mariadb.conf.d/50-server.cnf »** (L'affinement des IP de connexion s'étant fait au niveau de l'utilisateur).

bind-address = 0.0.0.0

Nous allons tester une connexion depuis le serveur Web :



Maintenant que nous avons réussi notre connexion à la base de données, nous allons pouvoir mettre en place un certificat SSL généré par notre autorité de certification sur l'OpnSense.
Le but de ce certificat étant de sécuriser les échanges entre le serveur web et la base de données.

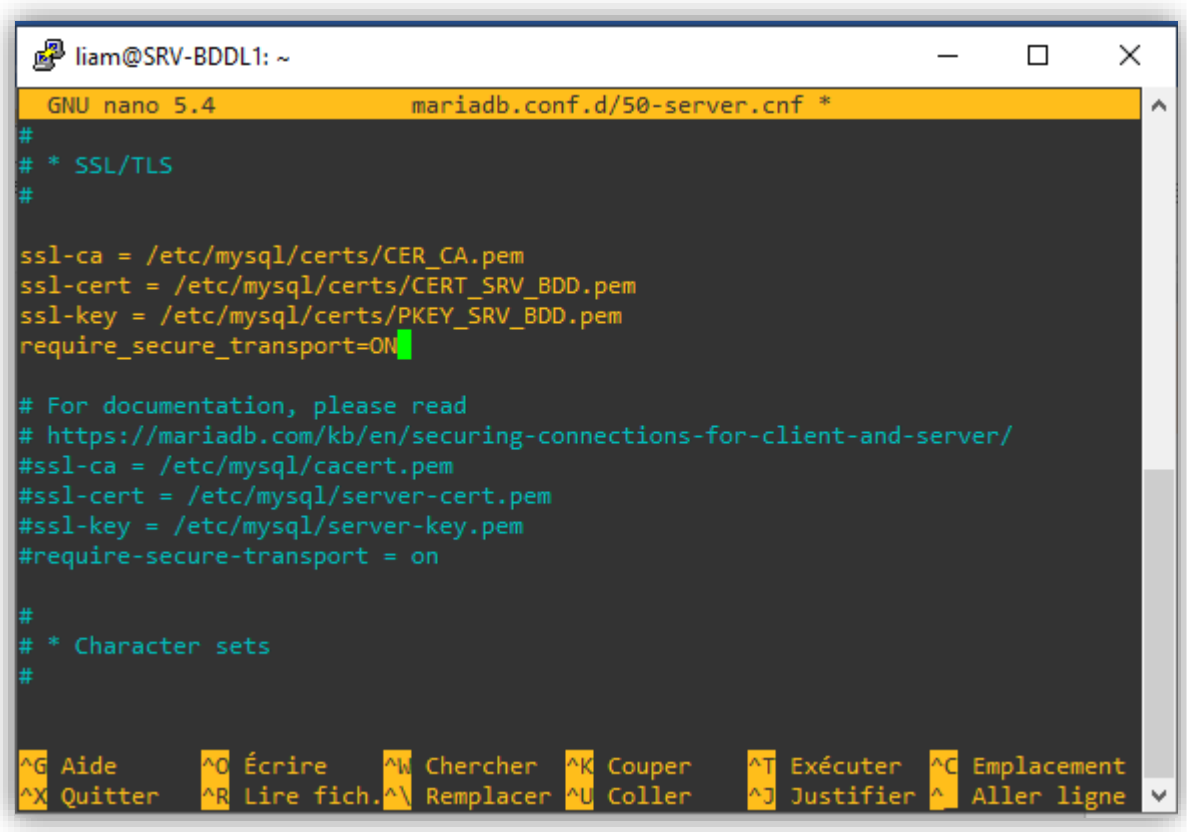
Nous allons donc sur le pare-feu pour générer le certificat serveur.

Ensuite, nous récupérons les informations suivantes :

- Clef privé du serveur
- Certificat du serveur
- Certificat de l'autorité de certification de notre Opnsense

Nous copions maintenant les certificats sur notre serveur de base de données dans le dossier **« /etc/mysql/certs ».**

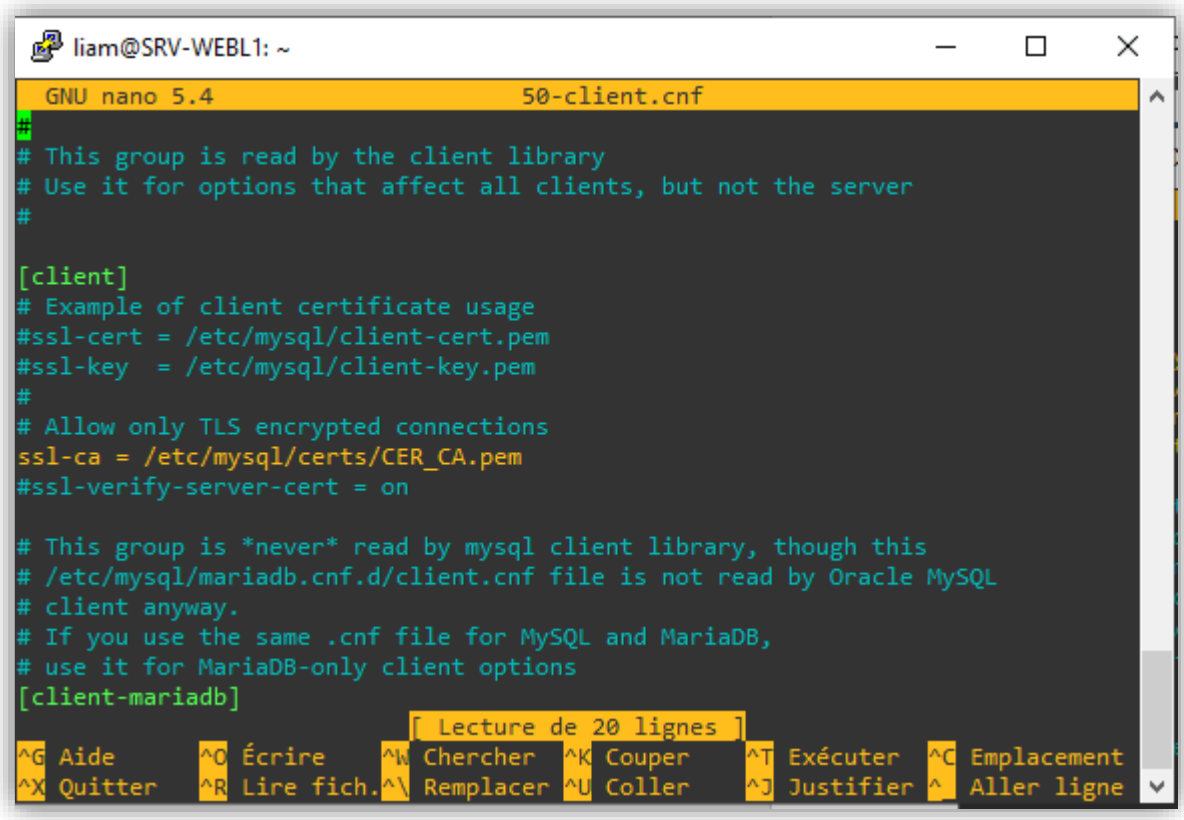
Une fois cela fait, nous allons indiquer l'emplacement des certificats et aussi activer l'utilisation de SSL.
Nous éditons donc le fichier de configuration **« mariadb.conf.d »**



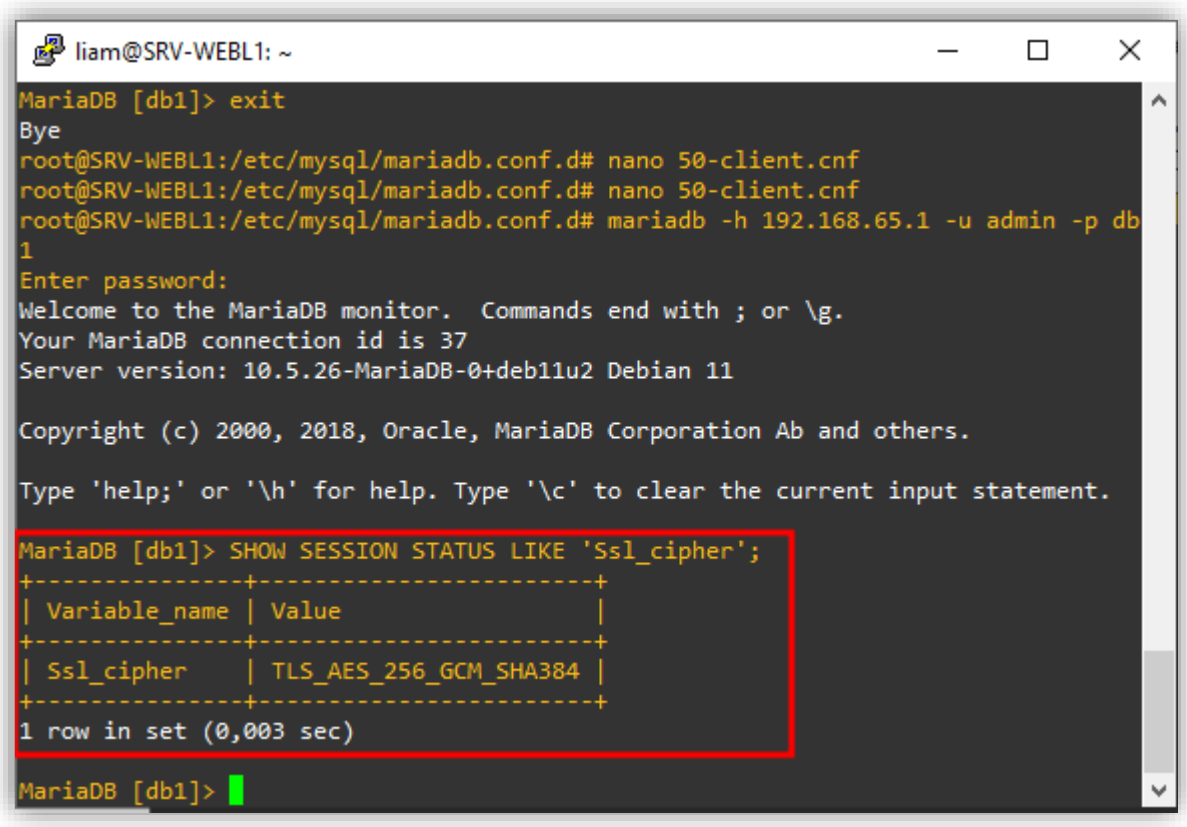
Il est aussi également nécessaire d'activer l'utilisation de SSL pour notre utilisateur via les commandes suivantes :

```
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'192.168.70.1' IDENTIFIED BY 'admin' REQUIRE
SSL;
FLUSH PRIVILEGES;
```

Nous allons désormais nous rendre sur notre serveur WEB et copions le certificat de l'autorité de certification.
Nous éditons le fichier de configuration du client MariaDB pour lui indiquer son emplacement :



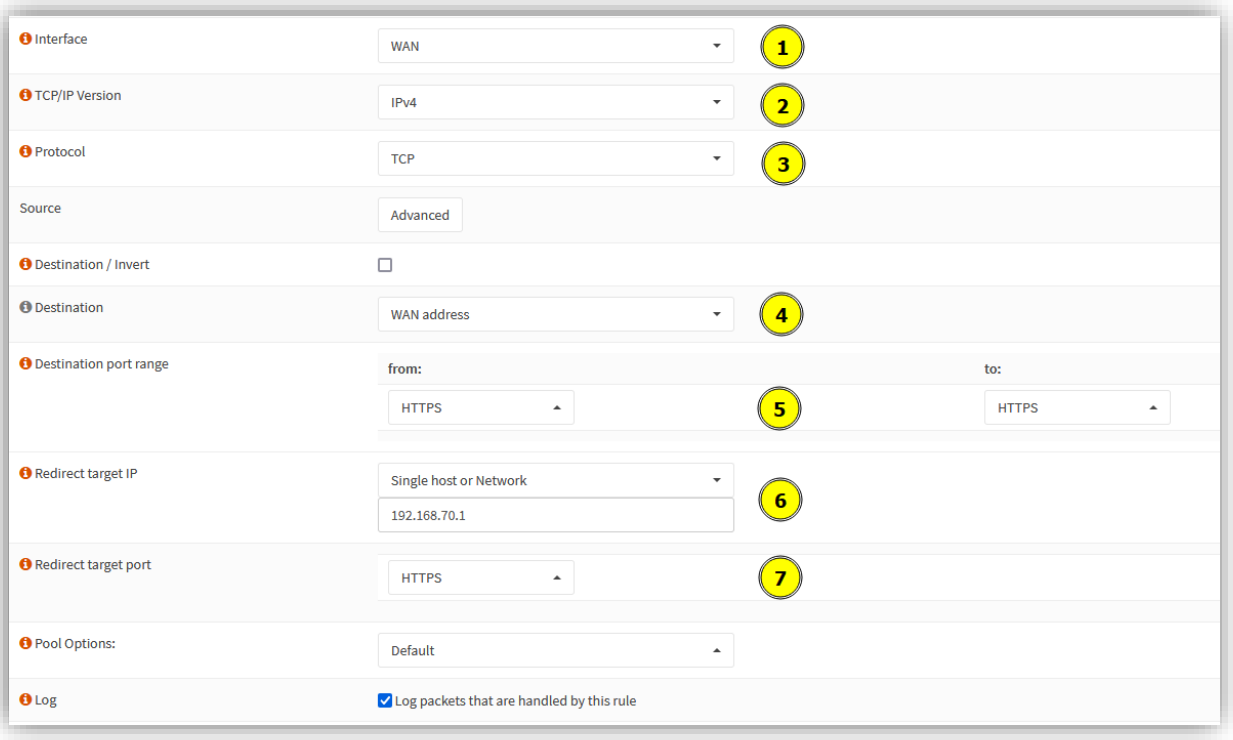
Une fois cela fait, nous pouvons tenter une connexion et vérifier l'utilisation de SSL :



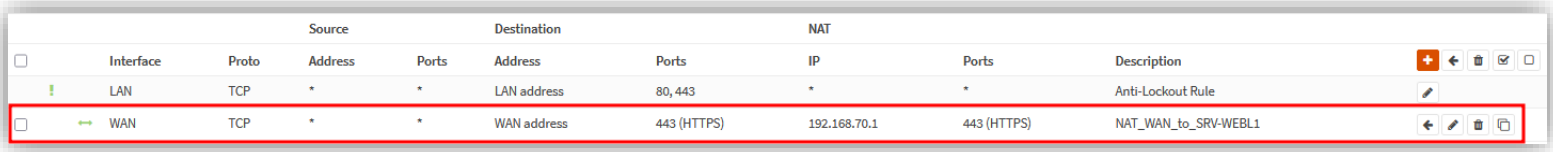
Maintenant, notre base de données est fonctionnelle et sécurisée.

2.4 Configuration de la règle NAT

Maintenant que nous avons configuré tous nos serveurs, nous allons créer une règle NAT qui va rediriger le trafic HTTPS arrivant sur notre interface WAN vers notre serveur WEB.
Nous allons dans le menu **« Firewall – NAT – Port Forward »** et cliquons sur **« add »**

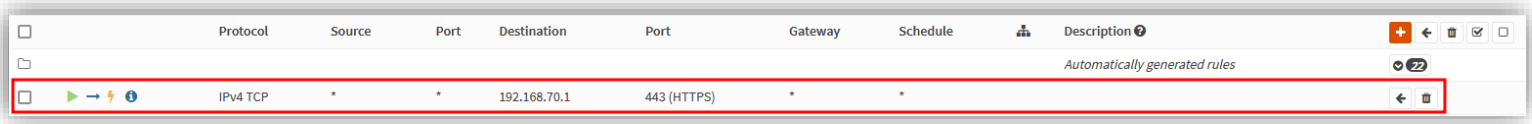


1. Interface d'arrivée du trafic
2. Protocole IPV4
3. Protocole TCP
4. Adresse de destination du flux
5. Port de destination
6. IP de redirection
7. Port de redirection

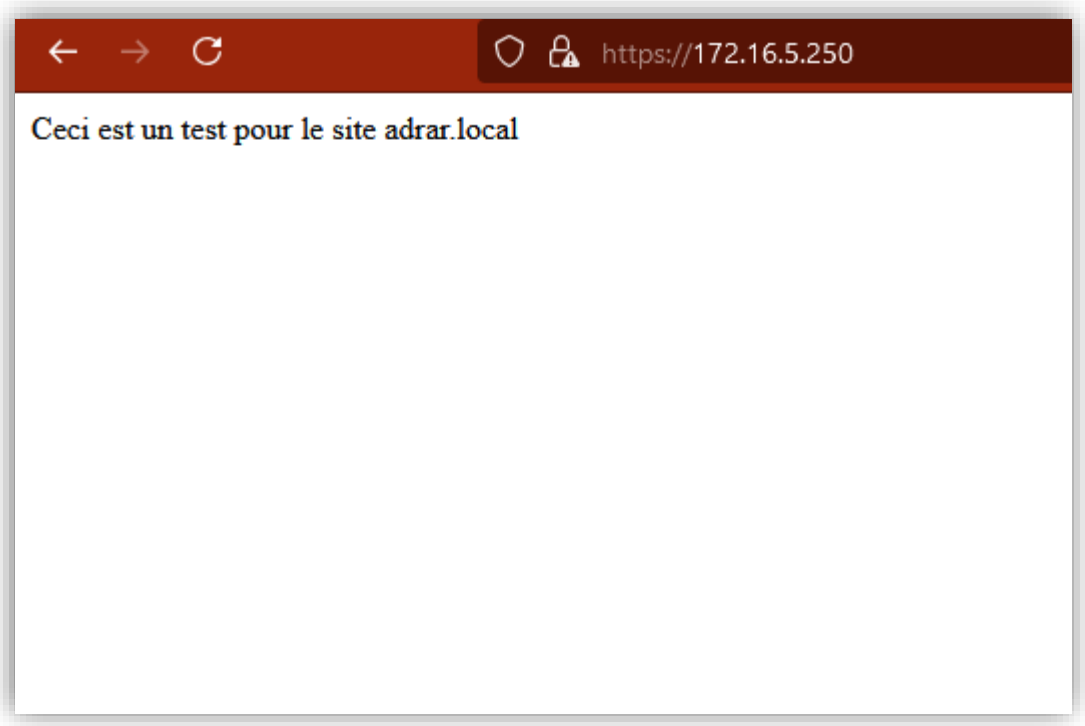


			Source		Destination		NAT		
	Interface	Proto	Address	Ports	Address	Ports	IP	Ports	Description
	LAN	TCP	*	*	LAN address	80, 443	*	*	Anti-Lockout Rule
	WAN	TCP	*	*	WAN address	443 (HTTPS)	192.168.70.1	443 (HTTPS)	NAT_WAN_to_SRV-WEBL1

Une règle de pare feu autorisant le trafic est générée automatiquement :



Puis nous testons notre accès au site Web via l'interface WAN :



Maintenant que cela est fait, nous allons créer les règles de pare feu pour le reste de nos interfaces.

2.3 Configuration des règles de pare feu

Pour toutes informations supplémentaires sur les règles en place, merci de vous référer au chapitre **1.2 Evolution de l'infrastructure**

Voici la configuration des interfaces ainsi que la vérification de leur bon fonctionnement (une seule règle est testée dans cette procédure, les autres vérifications suivent le même principe) :

DMZ

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description
								Automatically generated rules
	IPv4 TCP	DMZ net	*	*	443 (HTTPS)	*	*	Allow_HTTPS
	IPv4 UDP	DMZ net	*	SRV_DNS	53 (DNS)	*	*	Allow_DNS
	IPv4 TCP	DMZ net	*	SRV_BDDL1	3306	*	*	Allow_SQL

Interface	Time	Source	Destination	Proto	Label
DMZ	2024-10-20T17:21:09	192.168.70.1:49974	192.168.65.1:3306	tcp	Permit_DMZ_to_BDD

LAN

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description
								Automatically generated rules
	IPv4 TCP	LAN net	*	*	443 (HTTPS)	*	*	Allow_HTTPS
	IPv4 TCP	LAN net	*	*	80 (HTTP)	*	*	Allow_HTTP
	IPv4 UDP	LAN net	*	SRV_DNS	53 (DNS)	*	*	Allow_DNS
	IPv4 UDP	*	68	*	67	*	*	Allow_DHCP
	IPv4 TCP	LAN net	*	DMZ net	22 (SSH)	*	*	Allow_SSH_to_DMZ
	IPv4 TCP	LAN net	*	SRV net	22 (SSH)	*	*	Allow_SSH_to_SRV

SRV

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description
								Automatically generated rules
	IPv4 TCP	SRV net	*	*	443 (HTTPS)	*	*	Permit_SRV_HTTPS
	IPv4 UDP	SRV net	*	SRV_DNS	53 (DNS)	*	*	Permit_SRV_DNS

WAN

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description
								Automatically generated rules
	IPv4 TCP	*	*	192.168.70.1	443 (HTTPS)	*	*	
	IPv4 UDP	WAN net	*	SRV_DNS	53 (DNS)	*	*	Allow_DNS

3. Mise en place d'une DMZ externe avec Reverse Proxy

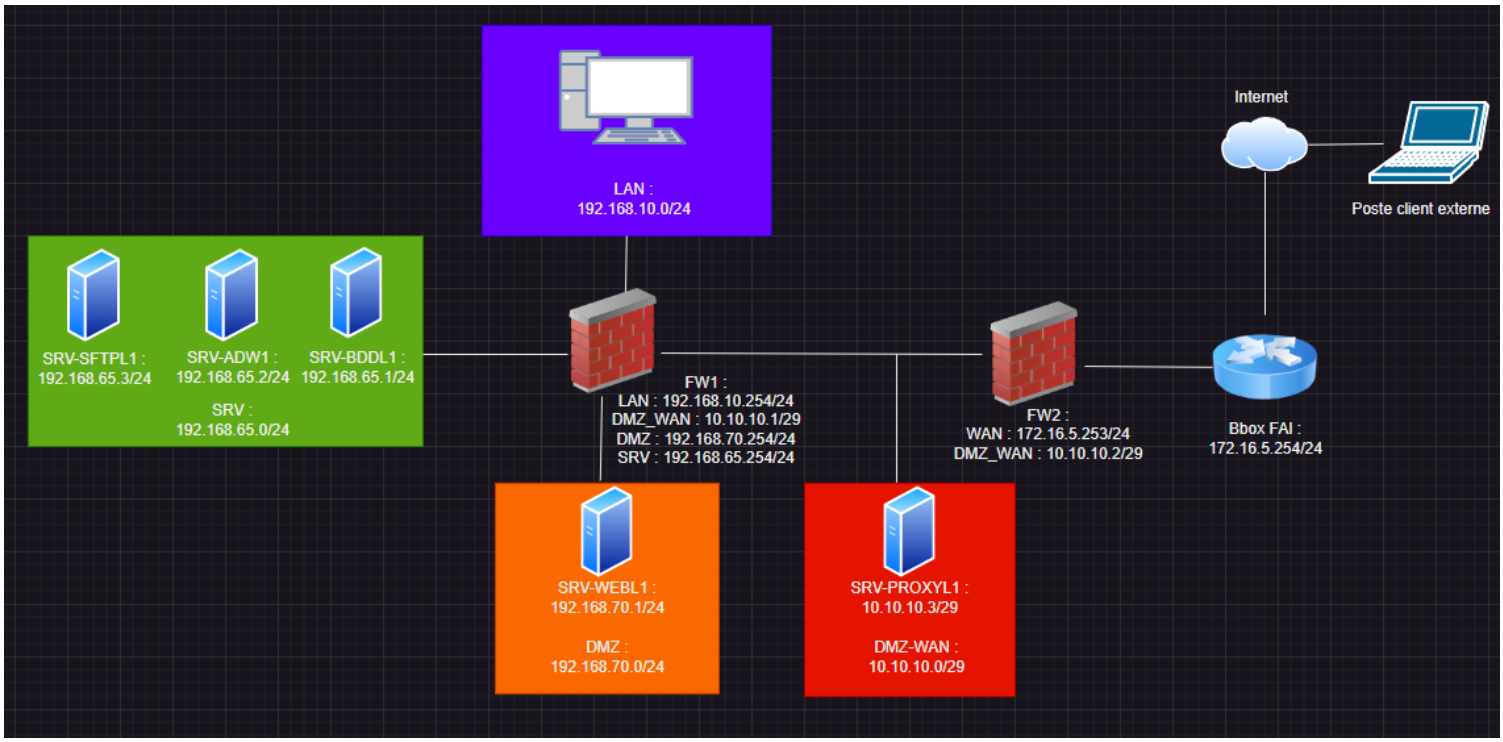
3.1 Evolution de l'infrastructure en suivant les recommandation ANSSI

Dans le cadre de l'amélioration de l'infrastructure au sein de l'ADRAR, la mise en place d'une DMZ externe ainsi que d'un reverse proxy pourrait être nécessaire.

En effet, selon les recommandations de l'ANSSI il est préférable de mettre en place une **« DMZ Externe ».** Pour cela, il faut ajouter un deuxième pare feu qui servira à un premier filtrage des flux arrivant sur le WAN. Le Reverse Proxy se chargera du rôle d'intermédiaire et renverra les requêtes sur les serveurs Web dans la **« DMZ Interne »**
Celui-ci sera connecté à une interface **« DMZ EXTERNE »** ou se situera le reverse proxy.

Le reverse proxy servira à repartir les charges entre les différents serveurs, gérer les négociations SSL/TSL, améliorer les performances via la mise en place d'un cache.
Il sera aussi utile afin d'anonymiser les serveurs de notre LAN en servant d'intermédiaire entre les requêtes HTTPS venant de l'extérieur et la réponse de notre serveur Web.

Voici le schéma de la solution :



Une interface nommée **« DMZ_WAN »** est donc ajoutée sur nos deux pare feu pour la DMZ externe avec un réseau en **10.10.10.0/29**

3.2 Evolution des règles de pare feu

Une nouvelle infrastructure nécessite donc des nouvelles règles de pare feu.

Les règles mises en place précédemment sur les interfaces **LAN**, **SRV**, **DMZ** du « **FW1** » resteront inchangées.

Voici les nouvelles règles mises en place :

1. **FW1** :

DMZ_WAN				
Protocol	IP Source	Port source	IP Destination	Port Destination
TCP	10.10.10.3/29	any	192.168.70.1/32	443

2. **FW2** :

WAN				
Protocol	IP Source	Port source	IP Destination	Port Destination
TCP	any	any	10.10.10.3/29	443
UDP	172.16.5.253/32	Any	192.168.65.2	53

DMZ_WAN				
Protocol	IP Source	Port source	IP Destination	Port Destination

Sur le **FW2** aucune règle n'est définie sur l'interface **« DMZ_WAN »** car l'autorisation de connexion de notre reverse proxy vers le serveur Web est faite sur le FW1. En effet, c'est directement sur ce pare feu que le trafic arrivera en entrée via le reverse proxy.

3.3 Mise en place de la solution

Pour le reverse Proxy nous avons choisi **« Haproxy »,** celui-ci offrant de bonnes performances, la possibilité de mise en place de haute disponibilité ainsi que du load-balancing entre les serveurs Web.

Voici les deux termes essentiels à connaitre lorsque l'on parle de reverse proxy :

- Frontend : C'est le serveur Reverse Proxy qui traitera les demandes d'accès au serveur Web venant de l'extérieur.
- Backend : Ce sont les serveurs Web dans notre LAN devant être joignables depuis l'extérieur.

Voici les prérequis avant de commencer la configuration de notre reverse Proxy :

- Modifier le **FW1** pour lui assigner une nouvelle interface qui servira pour la « **DMZ_WAN**» en **10.10.10.1/29**.
  Ajouter une nouvelle route par défaut sur cette interface vers le « **FW2** » qui redirigera les flux vers l'extérieur.
- Configurer le **FW2** avec 2 cartes réseau :
  - Une interface **WAN** en **172.16.5.253/24**
  - Une interface **DMZ_WAN** en **10.10.10.2/29**

Nous commençons maintenant la configuration du serveur **« Haproxy »** sous Debian 12.
Nous installons le paquet :

```
apt install haproxy
```

Une fois cela fait, il va être nécessaire d'ajouter une route statique pour joindre la DMZ Interne (la route par défaut du serveur étant le FW2 accédant au WAN).
Nous éditons donc le fichier **« /etc/network/interfaces »** et ajoutons la ligne :

```
up ip route add 192.168.70.0/24 via 10.10.10.1
```

Maintenant que cela est fait, il va être nécessaire d'ajouter un certificat pour le serveur qui sera généré par notre autorité de certification.
Il faut donc générer un fichier **« .pem »** qui contient le certificat serveur et la clé privée associée.

Nous copions le fichier nommé **« CERT_SRV-PROXYL1.pem »** dans le dossier **« /etc/haproxy/certs ».**
(Il est aussi nécessaire d'ajouter le certificat de l'autorité de certification pour la vérification des certificats SSL de nos backends)

```
root@SRV-PROXYL1:~# ls /etc/haproxy/certs/
CERT_CA.pem  CERT_SRV-PROXYL1.pem
```

Nous pouvons maintenant éditer le fichier de configuration HAProxy **« /etc/haproxy/haproxy.cfg »** :

```
# Frontend - Entrée de trafic
frontend https_front
    bind 10.10.10.3:443 ssl crt /etc/haproxy/certs/CERT_SRV-PROXYL1.pem
    default_backend servers_backend

# Backend - Serveurs en round-robin
backend servers_backend
    balance roundrobin
    server srv-webl1.adrar.local 192.168.70.1:443 ssl verify required ca-file /etc/haproxy/certs/CERT_CA.pem
```

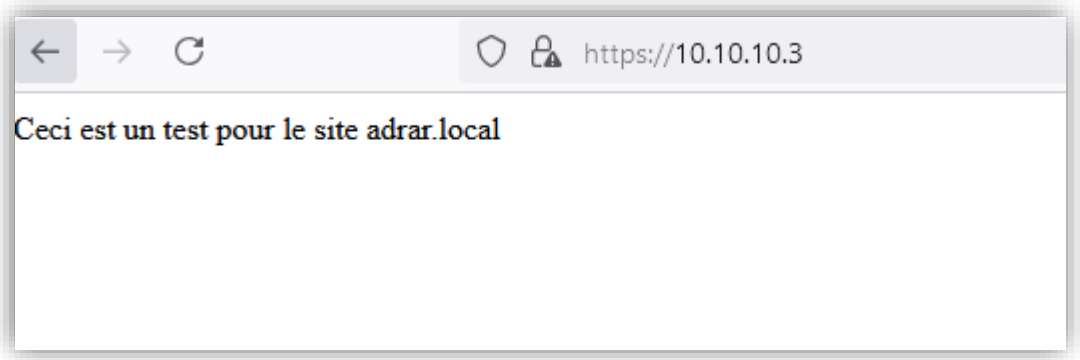
Plusieurs éléments sont indiqués dans ce fichier :

1. Nous définissons ici les configurations du serveur HAProxy avec le nom du **« Frontend »**
2. IP et port d'écoute du serveur ainsi que son certificat
3. Configuration du **« Backend »** avec le nom du **« pool de Backend ».**
   Le **« balance roundrobin »** est la partie configuration du Load Balancer pour répartir les charges (n'ayant qu'un serveur actuellement, celui-ci n'est pas utilisé mais reste configuré en cas d'évolution)
4. IP et port d'écoute de notre **« Backend »** ainsi que le certificat de la CA afin de vérifier son certificat.

Maintenant que notre configuration est faite, nous relançons le service :

```
systemctl restart haproxy
```

Nous allons pouvoir tester notre accès au site web en passant directement par le HAProxy :

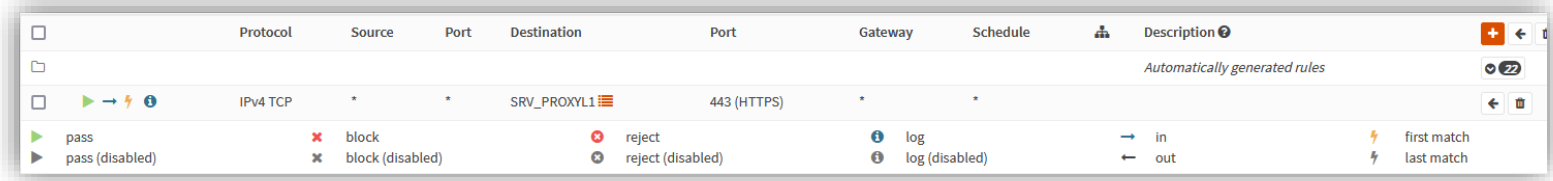


DMZ_LAN	→	2024-10-27T00:40:55	10.10.10.3:56892	192.168.70.1:443	tcp	Allow_SRV-PROXYL1_to_SRV-WEBL1

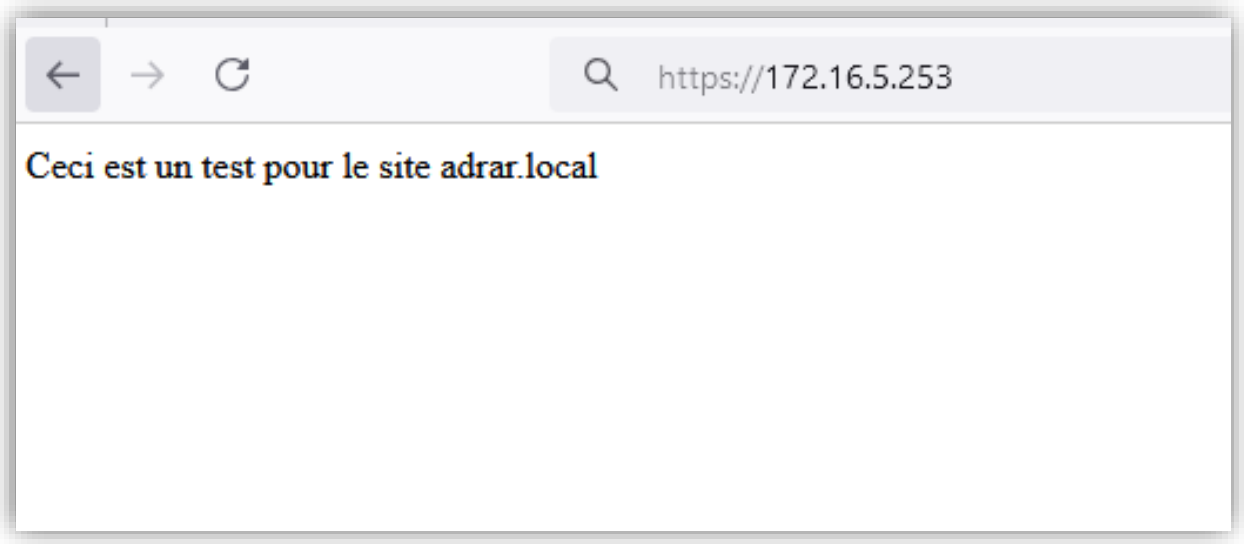
Maintenant que notre Reverse Proxy est fonctionnel, nous allons configurer une règle NAT qui redirige toutes les demandes HTTPS arrivant sur notre interface **WAN** vers notre **HAProxy** :

IP Cible	Port Cible	IP de redirection	Port de redirection
IP interface WAN : 172.16.5.253/32	Port HTTPS : TCP 443	IP Serveur WEB : 10.10.10.3/32	Port HTTPS : TCP 443

Une règle de pare feu est automatiquement générée pour autoriser les flux :



Nous allons maintenant effectuer notre test via notre interface **WAN** :



Tout est fonctionnel, nos règles ainsi que le reverse proxy.

4. Conclusion

En conclusion, ce projet a sécurisé l’infrastructure de l'ADRAR avec une DMZ pour le serveur Web et une base de données accessible en interne. L’ajout d’une DMZ externe avec un reverse proxy, suivant les recommandations de l’ANSSI, permet une sécurité renforcée avec un filtrage avancé et une optimisation des performances via le cache, tout en anonymisant les serveurs du LAN. Cette architecture répond aux besoins actuels et anticipe des améliorations futures pour une sécurité accrue.

5. Annexes

5.1 Documentations et références

Configuration du serveur Web Apache2 : https://debian-facile.org/utilisateurs:hypathie:tutos:creer-son-site-web-en-php-_-prendre-en-main-apache2-sur-jessie

Configuration du SSL sur le serveur Apache2 : https://www.it-connect.fr/configurer-le-ssl-avec-apache-2%EF%BB%BF/

Création de règles de pare feu Opnsense : https://www.zenarmor.com/docs/network-security-tutorials/how-to-configure-opnsense-firewall-rules

Génération des certificats via CA Opnsense : https://www.zenarmor.com/docs/network-security-tutorials/how-to-manage-certificates-on-opnsense

Configuration du serveur MariaDB : https://doc.fedora-fr.org/wiki/Installation_et_configuration_de_MariaDB

Configuration du SSL sur MariaDB : https://mariadb.com/kb/en/securing-connections-for-client-and-server/

Recommandation de l'ANSSI sur la mise en place d'une DMZ : https://cyber.gouv.fr/sites/default/files/2020/06/anssi-guide-passerelle_internet_securisee-v3.pdf