

DOCUMENTATION Technique - TiersLieux86

Par



Topologie générale

Bâtiment A : 3 étages

RDC : Salle en accès libre avec 25 postes et 1 imprimante.

Étage 1 : Salle de formation avec 20 postes, 1 imprimante et 1 ordinateur portable.

Étage 2 : Salle de réunion avec 1 poste, 1 téléphone IP et 1 borne d'accès Wi-Fi.

Bâtiment B : 3 étages + RDC

RDC : Baie technique - 2 switch empilés, firewall ASA, routeur WAN, accès Internet

Étages 1 & 2 : Bureaux avec 2x15 machines et borne d'accès Wi-Fi

Étage 3 : Salle de réunion avec 1 poste, 1 téléphone IP et 1 borne d'accès Wi-Fi

Équipements principaux utilisés :

switchs Cisco 2960-24TT empilés pour les baies

routeur Cisco-PT

Pare-feu ASA 5505

Serveurs-PT pour les DNS, DHCP, AD, NAS, mail, impression

AccessPoint-PT pour les bornes d'accès Wi-Fi

PC-PT / Laptop-PT / Printer-PT / Téléphone IP pour les équipements utilisateur.ices

Plan d'adressage :

Plage 10.2.0.0/24

Passerelle par défaut : 10.2.0.254 (pare-feu ASA → interface inside)

DNS : 10.2.0.1

DHCP pool TL86-LAN :

-IP de départ 10.2.0.100

-100 utilisateur.ices maximum

DMZ :

- Plage 172.16.2.0/24
- DNS 172.16.2.11
- Web 172.16.2.12
- Mail 172.16.2.13

Réseau externe :

ASA interface outside : 192.36.253.2

Routeur WAN :

- Lan ASA 192.36.253.1
- WAN cloud : 203.0.113.1

Convention de nomenclature :

Salle de formation	BAE1FORMATIONPxx
Salle en libre-service	BAERDCLIBREXPxx
Salle d'impression du RDC	BAERDCIMPxx
Salle des serveurs	CENTRAL_SRV_xxx
DMZ	DMZ_SRV_xxx
Firewall	CENTRAL_Firewall
WAN	CENTRAL_RouterWAN

Ce réseau a été conçu en suivant les contraintes du document référence. La topologie respecte la structure physique des bâtiments, et tous les équipements sont fonctionnels. Des tests de connectivité ont été réalisés pour valider l'infrastructure. Le projet a été réalisé avec méthode et rigueur, malgré des bugs et crashes de Packet Tracer, et devrait constituer un socle fiable pour tout déploiement.

———Les machines———

Nous avons quatres machines :

- Routeur : VyOS
- Contrôleur AD : Windows serveur 2019
- Machine type utilisateur.ice : Windows 10 Éducation édition

Machine	Rôle	IP	OS	Notes
VyOS	Routeur & NAT	10.0.2.x/25 192.168.10.1	VyOS	NAT vers le net externe, et LAN
Windows Server 2019	Contrôleur de domaine AD	192.168.10.10	Windows Server 2019	DNS, DHCP, AD et DS
Windows 10	Client	192.168.10.x	Windows 10 éducation	Membre du domaine
NAS	Server de stockage	192.168.10.200	TrueNAS	Placé dans la DMZ pour des raisons de sécurité

Windows Server 2019

Installé en mode "Core"

Configuré avec une IP statique pour être facilement retrouvé dans le réseau.

Installation des différents services grâce à PowerShell

DHCP avec une plage d'IPs allant de 192.168.10.50 et 192.168.10.100, et netmask de 255.255.255.0.

Passerelle en 192.168.10.1 vers la machine VyOS

Présence des utilisateur.ices sur le serveur :

```
New-ADUser -Name "Jean Dupont" -SamAccountName "jdupont" -UserPrincipalName "jdupont@tierslieux86.local"
-AccountPassword (ConvertTo-SecureString "TiersLieux86" -AsPlainText -Force) -Enabled $true -Path
"CN=Users,DC=tierslieux86,DC=local"
Add-ADGroupMember -Identity "Administration" -Members "jdupont"
```

Présence des groupes Administration, Adhérents et Esporting

Windows 10 Éducation Édition

Ajout au domaine par PowerShell, attribution d'une IP fixe.

Accès au dossier partagé présent sur l'AD, ajouté manuellement

Accès au Net grâce au routeur VyOS

PowerShell utilisé pour joindre au domaine :

```
Add-Computer -DomainName "tierslieux86.local" -Credential (Get-Credential) -Restart
```

VyOS

Routeur gérant la transition entre l'intranet et le Net externe.

Transposition des adresses internes vers l'extérieur et vice-versa

Arborescence de l'AD (incluant les clients *ValorElec*)

tierslieux86.local

- └─ ValorElec
 - └─ RechercheEtDeveloppement
 - └─ Utilisateurs R&D (comptes)
 - └─ Groupe G_RnD ← 10 comptes
 - └─ Commercial
 - └─ Utilisateurs Commerciaux
 - └─ Groupe G_Commercial ← 1 compte
 - └─ Direction
 - └─ Utilisateurs Direction
 - └─ Groupe G_Direction ← 8 comptes

Installation de programmes avec les installeurs *.msi* et les GPOs :

1. Copie du fichier *.msi* en question dans le répertoire SYSVOL du contrôleur de domaine (ici 192.168.10.10) (SYSVOL : SysVol\tierslieux86.local\scripts)

2. Création d'une GPO par le gestionnaire de domaine ;
Gestion des politiques de groupes

- └─ Forêt
 - └─ Domaines
 - └─ tierslieux86.local
 - └─ Politique par défaut
 - └─ Contrôleurs de domaine
 - └─ WinServ2019
 - └─ Administrateurs
 - └─ Ordinateurs
 - └─ installAcrobat ← Emplacement de la nouvelle GPO d'installation

3. Sélection de "nouveau paquet" puis du fichier *.msi* par son chemin absolu dans le répertoire de fichiers du contrôleur de domaine

4. Sélection de la méthode d'installation "Assignée" pour que l'installation se fasse à la prochaine connexion sur les machines

(Il est également possible de sélectionner l'option "ignorer la langue lors du déploiement" pour que l'installateur se réfère à la langue définie par le contrôleur de domaine.

—Les scripts—

Scripts Powershell de gestion :
Group-Manager.ps1

```
param(
    [string]$CSVPath = $(Throw "Veuillez entrer un fichier CSV valide,
plz.")
)

function Set-ADGroupMember {
    param (
        [Parameter(Mandatory=$True)]
        [string]$User,
        [string[]]$Groups
    )

    foreach ($group in $Groups) {
        Add-AdGroupMember -Identity $group -Members $User | Out-Null
    }
}

# Importe les données depuis le fichier CSV
$data = Import-Csv -Path $CSVPath

foreach ($row in $data) {
    $user = $row.User
    # Le script peut assigner jusqu'à 13 groupes à un.e utilisateur.ice,
    en partant du
    # principe que l'en-tête du fichier CSV est comme suit :
    # Group1,Group2[ ... ]Group11,Group12,Group13
    $groups = $row.[Group1], $row.[Group2], $row.[Group3], $row.[Group4],
    $row.[Group4], $row.[Group5], $row.[Group6], $row.[Group7], $row.[Group8],
    $row.[Group9], $row.[Group10], $row.[Group11], $row.[Group12], $row.
    [Group13]

    Set-ADGroupMember -User $user -Groups $groups
}
```

Create-users.ps1

```
Import-Module ActiveDirectory

# Chemin vers le CSV
$csvPath = "C:\temp\Users.csv"
$users = Import-Csv -Path $csvPath

# Tableau pour stocker les informations de cr dentiels (username:password)
$credentialsOutput = @()

foreach ($user in $users) {
    Write-Output "Cr ation de l'utilisateur $($user.SamAccountName)..."

    New-ADUser `
        -Name $user.Name `
        -SamAccountName $user.SamAccountName `
        -UserPrincipalName $user.UPN `
        -AccountPassword (ConvertTo-SecureString $user.Password -
AsPlainText -Force) `
        -Enabled $true `
        -Path $user.OU

    # Ajout de l'utilisateur au groupe s'il existe
    if ($user.Group -and (Get-ADGroup $user.Group -ErrorAction
SilentlyContinue)) {
        Add-ADGroupMember -Identity $user.Group -Members
$user.SamAccountName
    } else {
        Write-Warning "Le groupe $($user.Group) n'existe pas ou n'a pas
 t  trouv ."
    }

    # Ajouter la ligne de cr dentiels au tableau
    $credentialsOutput += "$($user.SamAccountName): $($user.Password)"
}

Write-Host "Import termin  !"

# D finir le chemin du document Word sur le bureau de l'utilisateur
courant
$desktopPath = [Environment]::GetFolderPath("Desktop")
$docPath = Join-Path $desktopPath "InfosLoginUtilisateur-ices.docx"

# Cr er une instance de Word via COM
$word = New-Object -ComObject Word.Application
$word.Visible = $false
$doc = $word.Documents.Add()

# Ins rer les cr dentiels dans le document Word
$range = $doc.Range()
foreach ($line in $credentialsOutput) {
    $range.InsertAfter("$line`r`n")
    # Mettre   jour le range   la fin du document
    $range = $doc.Range($doc.Content.End - 1, $doc.Content.End - 1)
}

# Sauvegarder le document et fermer Word
$doc.SaveAs([ref] $docPath)
$doc.Close()
$word.Quit()

Write-Host "Les identifiants ont  t  enregistr s dans : $docPath"
```

Manage-users.ps1

```
param(
    [Parameter(Mandatory=$true)]
    [string]$UserName,

    [Parameter(Mandatory=$true)]
    [ValidateSet("Add", "Remove")]
    [string]$Action,

    [Parameter(Mandatory=$true)]
    [string]$Role
)

# Charger le module ActiveDirectory
Import-Module ActiveDirectory

# Vérifier que l'utilisateur existe
$user = Get-ADUser -Identity $UserName -ErrorAction SilentlyContinue
if (-not $user) {
    Write-Error "L'utilisateur '$UserName' n'existe pas dans l'AD."
    exit 1
}

# Vérifier que le groupe existe
$group = Get-ADGroup -Identity $Role -ErrorAction SilentlyContinue
if (-not $group) {
    Write-Error "Le groupe '$Role' n'existe pas dans l'AD."
    exit 1
}

# Effectuer l'action demandée
if ($Action -eq "Add") {
    try {
        Add-ADGroupMember -Identity $Role -Members $UserName
        Write-Host "L'utilisateur.ice '$UserName' a été ajouté au groupe '$Role'."
    }
    catch {
        Write-Error "Erreur lors de l'ajout de '$UserName' au groupe '$Role' : $_"
    }
}
elseif ($Action -eq "Remove") {
    try {
        Remove-ADGroupMember -Identity $Role -Members $UserName -Confirm:
        $false
        Write-Host "L'utilisateur.ice '$UserName' a été retiré du groupe '$Role'."
    }
    catch {
        Write-Error "Erreur lors du retrait de '$UserName' du groupe '$Role' : $_"
    }
}
```


Create-OU.ps1

```
Import-Module ActiveDirectory

# Définition de la liste des OU à créer.
# Pour chaque OU, on définit :
# - Name : le nom de l'OU à créer
# - ParentPath : le conteneur (DN) dans lequel l'OU sera créée

$ouList = @(
    @{ Name = "VladEtec"; ParentPath = "DC=tierslieux86,DC=local" },
    @{ Name = "RechercheEtDeveloppement"; ParentPath =
"OU=VladEtec,DC=tierslieux86,DC=local" },
    @{ Name = "Commercial"; ParentPath =
"OU=VladEtec,DC=tierslieux86,DC=local" },
    @{ Name = "Direction"; ParentPath =
"OU=VladEtec,DC=tierslieux86,DC=local" }
)

foreach ($ou in $ouList) {
    $name = $ou.Name
    $parentPath = $ou.ParentPath

    # Vérifier si l'OU existe déjà dans le parent spécifié
    $exists = Get-ADOrganizationalUnit -Filter "Name -eq '$name'" -
SearchBase $parentPath -ErrorAction SilentlyContinue

    if ($exists) {
        Write-Host "L'OU '$name' existe déjà dans '$parentPath'."
    }
    else {
        Write-Host "Création de l'OU '$name' dans '$parentPath'..."
        New-ADOrganizationalUnit -Name $name -Path $parentPath
    }
}

Write-Host "Création des OU terminée !"
```

Gestion de la sécurité et application générale :

Mots de passe :

Set-ADAccountPasswordPolicy -MinimumPasswordLength 8 # Taille minimum

Set-ADAccountPasswordPolicy -MaximumPasswordAge "30" # Durée minimum de
non-réutilisation

Set-ADAccountPasswordPolicy -EnforcePasswordHistory 12 # Historique des mots
de passe sauvegardés

Set-ADA

Suppression du panneau de configuration pour les utilisateurs non
authentifiés :

Création d'une GPO sous "Configuration des machines" > "Politiques" >

"Templates administratifs" > " Menu de démarrage et barre des tâches" >

Activer la politique nommée "Remove Control Panel applet from the start menu"