Documentation Technique

Rédigée par
L'équipe VirtualLaunch

HISTORIQUE DES RÉVISIONS

Date	Version	Description	Auteur	Relecteur
16/11/2022	1.0	Création de la documentation technique	Matéo MIOSSEC Benoît PREVOST	Équipe
16/11/2022	2.0	Mise à jour des scripts existant et ajout du script Lifetime.sh	Matéo MIOSSEC	Équipe
11/01/2023	3.0	Mise en place du VPN	Arthur MERLE	Équipe
23/01/23	3.1	Modification / correction	Dany BRUNELLO Aksel GRELET Matéo MIOSSEC	Équipe

TABLES DES MATIÈRES

INSTALLATION DES VMS 4
HOSTSCRIPT.SH 4
GUESTSCRIPT.SH 7
LIFETIME.SH 8
INSTALL.PS1 9
MYRTILLE 10
QU'EST-CE QUE C'EST 10
INSTALLATION - DOCKER 10
ARBORESCENCE 11
ACTIVE DIRECTORY 14
VPN SSL 16
CONFIGURATION D'UN TUNNEL 16
DROIT D'ACCÈS VPN SSL 17
RÈGLE DE FILTRAGE IMPLICITE POUR LE VPN SSL 18
CONFIGURATION DU SERVICE VPN SSL 19
FILTRAGE ET NAT 21

INSTALLATION DES VMS

HostScript.sh

Dans un premier temps, nous pouvons y retrouver une liste de variables :

- "ID" correspond à l'identifiant de la machine.
- "mdp" est le mot de passe de l'utilisateur d'origine des machines virtuelles sous Linux.
- "user" est l'identifiant d'origine des machines virtuelles sous Linux.
- "mdpW" est le mot de passe de l'utilisateur d'origine des machines virtuelles sous Windows.
- "userW" est l'identifiant d'origine des machines virtuelles sous Windows.

```
ID="$2"
mdp="ubuntu"
user="ubuntu"
mdpW="windows"
userW="windows"
```

Nous avons écrit une fonction qui permet de vérifier si le processus est bien terminé avant de pouvoir lancer la commande suivante

```
function Ntime {
    #Si le processus ayant pour nom "$prog" est actif Sur la VM on attend
    while [ "$(vmrun -T ws -gu $user -gp $mdp listProcessesInGuest /home/gitaxias/Documents/VMware/"$ID"/
    "$ID".vmx | grep "$prog" | cut -f1)" != "" ]
    do
        sleep 1
    done
}
```

En fonction des arguments, on interagit avec les VMs.

Dans cette première condition, nous vérifions si l'argument donné est "start", si c'est le cas cela lance la machine virtuelle.

```
#En fonction des arguments{start, stop, delete ...} on interagit avec les VMs.
if [[ "$1" = "start" ]] ; then
    #On démarre la vm.
    vmrun -T ws start /home/ubuntu/VMware/"$ID"/"$ID".vmx nogui
    echo "VM démarrée"
```

Si la condition précédente n'est pas remplie, nous vérifions si l'argument est "clone".
Dans ce cas, nous créons un dossier pour la nouvelle machine, puis nous lançons la commande de clonage de la machine d'origine.
Le clonage se fait en fonction du système d'exploitation et on récupère l'identifiant du processus.

```
elif [[ "$1" = "clone" ]] ; then
    mkdir /home/ubuntu/VMware/"$ID"
    vmrun -T ws clone /home/ubuntu/VM_Origin/Ubuntu_origin/"$3"_origin.vmx /home/ubuntu/VMware/"$ID"/"$ID".vmx
    -cloneName="$ID"  linked & VMPid=$!
    echo "le pid de la vm est $VMPid"
    sleep 5
```

Nous modifions ensuite la RAM et le CPU de la machine virtuelle puis on la démarre.

```
lineMem=$(sed -n '/memsize/p' /home/ubuntu/VMware/"$ID"/"$ID".vmx)
lineCpu=$(sed -n '/numvcpus/p' /home/ubuntu/VMware/"$ID"/"$ID".vmx)
if [[ "$lineMem" != *"$6"* ]] ; then
    sed -i "s/$lineMem/memsize = \"$7\"/g" /home/ubuntu/VMware/"$ID"/"$ID".vmx
fi
if [[ "$lineCpu" != *"$7"* ]] ; then
    sed -i "s/$lineCpu/numvcpus = \"$7\"/g" /home/ubuntu/VMware/"$ID"/"$ID".vmx
fi
vmrun -T ws start /home/ubuntu/VMware/"$ID"/"$ID".vmx nogui
echo "VM démarrée"
```

Si le système d'exploitation qui a été cloné est un Windows, nous créons un répertoire dans lequel nous y mettons la liste des programmes ainsi que le script qui permettra de les installer. Puis nous exécutons ce script.

```
#En fonction de l'OS on exécute les commandes ci-dessous.
if [[ "$3" == "Windows" ]] ; then
    #On crée un Directory dans la VM
    vmrun -T ws -gu $userW -gp $mdpW createDirectoryInGuest  /home/ubuntu/VMware/"$ID"/"$ID".vmx C:'\'Program
    Files'\'ScriptInstall
    echo "directory created"

    #On envoie sur la VM la liste des programmes à installer.
    vmrun -T ws -gu $user -gp $mdp CopyFileFromHostToGuest /home/ubuntu/VMware/"$ID"/"$ID".vmx /home/ubuntu/GuestScript.sh
    C:'\'Program Files'\'ScriptInstallGuestScript.sh
    echo "programs list copyied"

    #On envoie sur la VM le script d'installation des programmes.
    vmrun -T ws -gu $user -gp $mdp CopyFileFromHostToGuest /home/ubuntu/VMware/"$ID"/"$ID".vmx /home/ubuntu/"$4" C:'\'Program
    Files'\'ScriptInstall'\'"$4"
    echo "install script copyied"

    #On exécute le script dans la VM.
    vmrun -gu $user -gp $mdp runScriptInGuest /home/ubuntu/VMware/"$ID"/"$ID".vmx -interactive "" "C:'\'perl'\'perl.exe
    C:'\'Program Files'\'$4"
    echo "script launched and has ended without errors"
```

Sinon on refait les mêmes étapes pour Linux

Sinon, si le système d'exploitation est un système Linux. Nous faisons les mêmes manipulations, cependant nous précisons la suppression de certains fichiers.

```
    else
        echo "VM started"
        vmrun -T ws -gu $user -gp $mdp createDirectoryInGuest  /home/gitaxias/Documents/VMware/"$ID"/"$ID".
        vmx /home/"$user"/ScriptInstall
        echo "directory created"
        vmrun -T ws -gu $user -gp $mdp CopyFileFromHostToGuest /home/gitaxias/Documents/VMware/"$ID"/"$ID".
        vmx /home/gitaxias/Documents/ProjetPi/"$5" /home/"$user"/ScriptInstall/"$5"
        echo "programs list copyied"
        vmrun -T ws -gu $user -gp $mdp CopyFileFromHostToGuest /home/gitaxias/Documents/VMware/"$ID"/"$ID".
        vmx /home/gitaxias/Documents/ProjetPi/"$4" /home/"$user"/ScriptInstall/"$4"
        echo "install script copyied"
        vmrun -T ws -gu $user -gp $mdp runScriptInGuest /home/gitaxias/Documents/VMware/"$ID"/"$ID".vmx  ""
        "/bin/sh /home/ubuntu/ScriptInstall/GuestScript.sh"
        echo "script launched and has ended without errors"
        vmrun -T ws -gu $user -gp $mdp deleteFileInGuest /home/gitaxias/Documents/VMware/"$ID"/"$ID".vmx /
        home/"$user"/ScriptInstall/"$4"
        vmrun -T ws -gu $user -gp $mdp deleteFileInGuest /home/gitaxias/Documents/VMware/"$ID"/"$ID".vmx /
        home/"$user"/ScriptInstall/"$5"
    fi
```

Sinon, si l'argument donné est "stop", nous éteignons la machine virtuelle.

```
elif [[ "$1" = "stop" ]] ; then
    vmrun -T ws stop /home/gitaxias/Documents/VMware/"$ID"/"$ID".vmx
```

Sinon, si l'argument donné est "delete", nous éteignons la machine et effaçons les fichiers.

```
elif [[ "$1" = "delete" ]] ; then

    vmrun -T ws stop /home/gitaxias/Documents/VMware/"$ID"/"$ID".vmx
    sleep 3
    rm -rf /home/gitaxias/Documents/VMware/"$ID"
```

Si la commande contient l'argument "-h" ou "help", nous affichons une aide.

```
elif [ "$1" = "-h" ] || [ "$1" = "--help" ] ; then
    echo "
    ScriptHost 2.0 (https://notresite/help)

    Usage : bash scriptVm.sh [Option] [Argument]

    !!! L'ordre des arguments est à respecter !!!

    Option:
        -start [ID] : démarre un VM déjà existante
        -clone [ID] [OS] [FichierProgramme] [ScriptProgramme]: crée un clone d'une vm mère ----- [User]-> on propose à la personne
        de créer un user sur le guest (pas encore implémentée)
        -stop [ID] : met en pause une vm déjà existante
        -delete [ID] : efface une vm
        -is_up [ID] : regarde si la vm est allumée
        -help : affiche cette aide"
fi
```

GuestScript.sh

Dans un premier temps, nous créons un fichier de logs afin de savoir quels programmes sont à jour ou non.

```
$non_installed_packets : ""
touch /home/ubuntu/ScriptInstall/log.txt
file="/home/ubuntu/ScriptInstall/$1"
echo $file >> /home/ubuntu/ScriptInstall/log.txt
```

On vérifie si le système est Fedora, Centos, Debian ou Kali, en fonction du système on exécute la partie correspondante, tant qu'il y a des lignes à lire, nous listons les programmes à installer dans le fichier logs et on installe les programmes du fichier programs.txt sinon on affiche que le script ne marche que sur les systèmes d'exploitation précédemment cités.

```
# Détecter le système d'exploitation parmi Fedora, Centos, ubuntu et kali
if [ -f /etc/fedora-release ]; then
    dnf update -y
    # Pour Fedora, installer les paquets présents dans le fichier programs.txt
    for i in $(cat $1); do
        dnf install $i -y
        if [ $? -ne 0 ]; then
            $non_installed_packets += $i
        fi
    done
elif [ -f /etc/centos-release ]; then
    cd /etc/yum.repos.d/
    sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
    sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
    # reviens dans le répertoir courant
    cd -
    #Mettre à jour le système
    yum update -y
    # Pour Centos, installer les paquets présents dans le fichier programs.txt
    for i in $(cat $1); do
        yum install $i -y
        if [ $? -ne 0 ]; then
            $non_installed_packets += $i
        fi
    done
elif [ -f /etc/debian_version ]; then
    echo "Updating" >> /home/ubuntu/ScriptInstall/log.txt
    #Mettre à jour le système
    sudo apt update
    sudo timedatectl set-timezone Europe/Paris

    echo "Updated, installing programs" >> /home/ubuntu/ScriptInstall/log.txt
    # Pour Ubuntu, installer les paquets présents dans le fichier programs_*.txt
    while read ligne || [ -n "$ligne" ]; do
        if test "$ligne" = "wireshark"; then
            sudo add-apt-repository ppa:wireshark-dev/stable -y
            sudo apt update
        elif test "$ligne" = "docker-ce"; then
            sudo apt install apt-transport-https ca-certificates curl software-properties-common
            curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
            sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
            sudo apt update
        fi
        sudo apt install $ligne -y
        sleep 1
        if [ $? -ne 0 ]; then
            $non_installed_packets += $i
        fi
        echo "Le programme $ligne a été installé" >> /home/ubuntu/ScriptInstall/log.txt
    done < $file
```

```
elif [ -f /etc/kali_version ]; then
    #Mettre à jour le système
    apt update
    # Pour Kali, installer les paquets présents dans le fichier programs.txt
    while read ligne || [ -n "$ligne" ]; do
    if [ $ligne == "wireshark" ]; then
            sudo add-apt-repository ppa:wireshark-dev/stable -y
            sudo apt update
        elif [ $ligne == "docker-ce" ]; then
            sudo apt install apt-transport-https ca-certificates curl software-properties-common
            curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
            sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
            sudo apt update
        fi
        sudo apt install $ligne -y
        sleep 1
        if [ $? -ne 0 ]; then
            $non_installed_packets += $i
        fi
        echo "Le programme $ligne a été installé" >> /home/ubuntu/ScriptInstall/log.txt
    done < $file
else
    echo "Ce script ne fonctionne que sur Fedora, Centos, Ubuntu et Kali" >> /home/ubuntu/ScriptInstall/log.txt
fi
```

Lifetime.sh

Ce script a pour but de vérifier et gérer la durée de vie des VMs en fonction de l'identifiant de la machine virtuelle.
Ce script s'exécutera toutes les minutes.
Les identifiants et leurs durées de vie respectives seront stockés dans une base de données.
La durée de vie, est exprimée en heures.

```
Standard = 72

life=$2
pid=$1

if [[ "$2" == "none" ]] ; then
    $life=$Standard
fi

afk=0

while [ $afk -lt $((($life*3600)/10)) ]
do
    if [[ $(top | grep "$pid") != "" ]]
    then
        $afk=0
    else
        $afk=$(($afk+1))
    fi
    sleep 10
done

vmrun -T ws stop /home/gitaxias/Documents/VMware/"$ID"/"$ID".vmx
sleep 3
rm -rf /home/gitaxias/Documents/VMware/"$ID"
```

install.ps1

Tant que le fichier avec la liste des programmes n'existe pas nous attendons

```
while(!(Test-Path "C:\Users\Etudiant\Desktop\UserInstall.txt")){
    Write-Output "Waiting for file"
    Start-Sleep 10
    }
```

Une fois le fichier trouvé, nous lisons le fichier ligne par ligne et nous installons chacun des logiciels.

```
foreach($line in Get-Content "C:\Users\Etudiant\Desktop\UserInstall.txt") {
    choco install $line -y
}
```

Nous supprimons ensuite le fichier ainsi que le script PowerShell.

```
Remove-Item "C:\Users\Etudiant\Desktop\UserInstall.txt"
Remove-Item "C:\Users\Etudiant\Desktop\install.ps1"
```

MYRTILLE

QU'EST CE QUE C'EST ?

Myrtille est un système permettant de rendre possible la virtualisation des postes de travail et des applications dans un navigateur Web uniquement en utilisant des technologies Web natives.

INSTALLATION - DOCKER

Myrtille est disponible sous forme d'image docker Windows.

Il faut tout d'abord installer une VM Windows avec Docker Desktop et activer les conteneurs Windows.

Puis on récupère l'image de Docker Hub avec la commande suivante :

```
docker pull cedrozor/myrtille
```

On exécute l'image en mode détaché et on redirige les ports 80 et 443 pour avoir accès à l'interface graphique.

```
docker run -d -p 80:80 -p 443:443 cedrozor/myrtille
```

Pour lister les conteneurs :

```
docker ps -a
```

Pour ouvrir un shell dans un conteneur (et pouvoir l'explorer, vérifier son adresse IP, ses logs, etc.)

```
docker exec -it <container ID> cmd
docker exec -it <container ID> powershell
```

Pour arrêter un conteneur :

```
docker stop <container ID>
```

ARBORESCENCE

Le site suit une arborescence toute particulière pour faciliter sa compréhension :

- Config: Ce dossier contient des fichiers de configuration pour le site web, tels que des constantes pour les informations de connexion à la base de données, les adresses IP, etc. Ces informations sont cruciales pour le fonctionnement du site web, il est donc important de les protéger contre les accès non autorisés.

- DAO: Ce dossier contient les différentes fonctions utilisées sur le site pour gérer les accès aux données. Ces fonctions peuvent inclure des requêtes pour récupérer des données de la base de données, des méthodes pour insérer ou mettre à jour des données, etc.

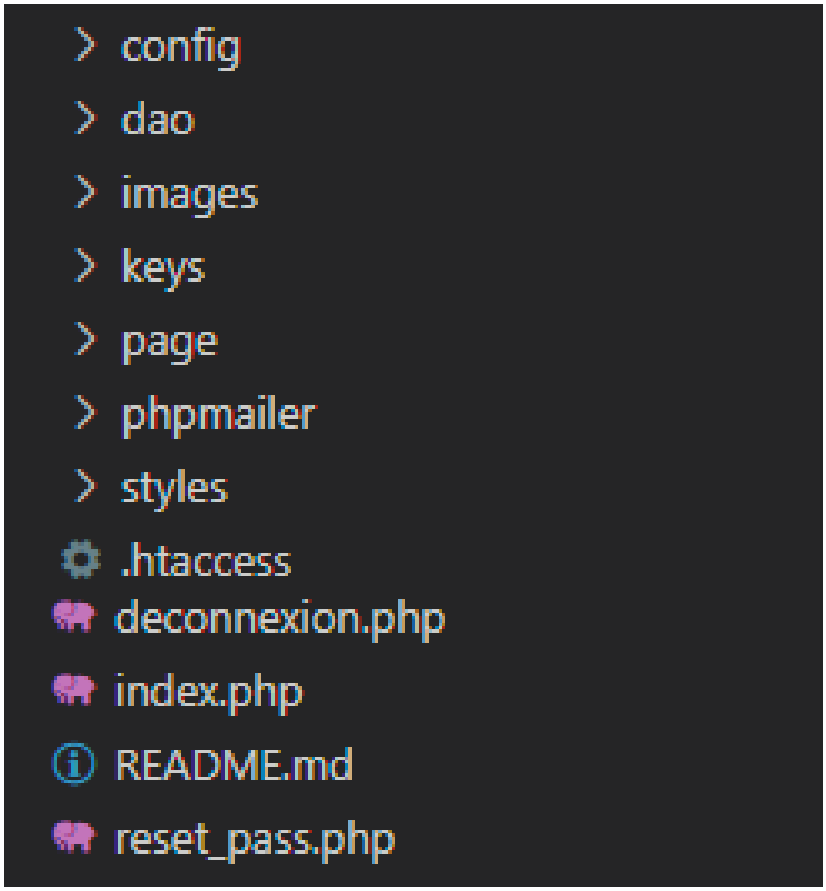
- Images: Ce dossier contient toutes les images utilisées sur le site, comme des images de fond, des images pour les boutons, etc. Il est important de s'assurer que les images sont bien organisées et nommées pour faciliter la gestion des images.

- Keys: Ce dossier contient des clés SSH utilisées pour la création de machines virtuelles. Il est important de s'assurer que ces fichiers ne sont pas exposés publiquement car cela pourrait entraîner des problèmes de sécurité.

- Pages: Ce dossier contient les pages du site, organisées en sous-dossiers pour les différentes catégories d'utilisateurs tels que administrateur, professeur et étudiant. Il peut inclure des fichiers pour les pages de contenu spécifiques, des scripts de traitement de formulaire, etc.

- Phpmailer: Ce dossier contient les fichiers pour l'extension PHP Mailer qui permet de gérer l'envoi de courriels via le site web, notamment pour les réinitialisations de mots de passe des utilisateurs.

- Styles: Ce dossier contient toutes les feuilles de style CSS utilisées pour styliser les pages du site web, avec des fichiers pour les différents éléments de la page (header, footer, etc) ou pour les différents type de pages (admin, user, ...) Il peut également inclure des fichiers JavaScript pour les interactions utilisateurs.



Nous avons ensuite une page :

- "index", Il s'agit de la page d'accueil du site. C'est la première page que les utilisateurs voient lorsqu'ils visitent le site.
- "déconnexion" : Il s'agit de la page qui gère la déconnexion de l'utilisateur. Il inclut des fonctions pour détruire les sessions utilisateur et rediriger vers la page de connexion.
- "reset_pass" : Il s'agit de la page qui gère le processus de réinitialisation de mot de passe. Il inclut notamment une fonction d'envoi de courriel avec des instructions pour réinitialiser le mot de passe.
- "htaccess" : Il s'agit d'un fichier de configuration pour le serveur web Apache qui permet de gérer les erreurs et les permissions d'accès à certains fichiers. Il peut inclure des redirections pour des pages spécifiques, des règles pour protéger les dossiers sensibles, des règles pour protéger les fichiers spécifiques, etc.

CONFIG.PHP

```php
<?php
    // Constantes pour la bd
    const DBS_HOST = 'localhost';
    const DBS_BASE = 'virtuallaunch';
    const DBS_USER = 'root';
    const DBS_PASS = '';

    // Constantes pour le SSH
    const SSH_IP = '137.74.94.64';
    const SSH_USER = 'ubuntu';
    const SSH_PASS = 'GJgTbd5z8fXG6A4q';

    // Constantes pour l'AD
    const AD_IP = '10.0.0.18';
    const AD_PORT = '389';
    const AD_SSH_PORT = '22';
    const AD_SUFFIX = '@virtuallaunch.local';
    const AD_ADMIN_USER = 'Administrateur';
    const AD_ADMIN_PASS = 'Virtuallaunch47!';
    const LDAP_TREE = 'CN=Users,DC=virtuallaunch,DC=local';
    const LDAP_TREE_ONLY_DC = 'DC=virtuallaunch,DC=local';

    // Constantes pour le mail
    const MAIL_HOST = 'mail.hyxoheberg.fr';
    const MAIL_USER = 'virtuallaunch@hyxo.fr';
    const MAIL_PASS = 'virtuallaunch47#';
?>
```

Le fichier config.php comporte les différentes informations de configuration nécessaires au projet. Nous y retrouvons:

- Les informations relatives à la Base De Données
- Les informations relatives à la connexion SSH
- Les informations relative à l'Active Directoy
- Les informations relatives au service mail (actuellement le serveur personnel de Dany Brunello, qu'il faudra à l'avenir, dans l'idéal, ne plus utiliser pour gérer son propre service)

ACTIVE DIRECTORY

Prérequis :

- Serveur sous Windows Server (Version 2022 sans interface graphique (préconisé))

Après l'installation rendez-vous dans le terminal, puis exécutez le script suivant :

```
# Demande à l'utilisateur de renseigner un nom de domaine et un TLD
$Domain = Read-Host "Merci d'entrer un nom de domaine sous la forme 'domain.tld'"
# Formate le nom de domaine afin de configurer le Netbios
$DomainCaps = $Domain.Split('.')[0]
$DomainCaps = $DomainCaps.ToUpper()
# Installation du service AD
Add-WindowsFeature AD-Domain-Services
# Configuration du service AD.
Import-Module ADDSDeployment
Install-ADDSForest `
-CreateDnsDelegation:$false `
-DatabasePath "C:\Windows\NTDS" `
-DomainMode "WinThreshold" `
-DomainName "$Domain" `
-DomainNetbiosName "$DomainCaps" `
-ForestMode "WinThreshold" `
-InstallDns:$true `
-LogPath "C:\Windows\NTDS" `
-NoRebootOnCompletion:$false `
-SysvolPath "C:\Windows\SYSVOL" `
-Force:$true
```

Ce script utilise PowerShell pour configurer un service Active Directory (AD). Il demande d'abord à l'utilisateur de saisir un nom de domaine et un TLD (domaine de premier niveau) sous la forme "domain.tld" à l'aide de la commande Read-Host.

Il utilise ensuite la commande Split pour séparer le nom de domaine en deux parties : le nom de domaine proprement dit et le TLD. Il met ensuite en majuscule le nom de domaine proprement dit à l'aide de la commande ToUpper.

Ensuite, il utilise la commande Add-WindowsFeature pour installer le service AD-Domain-Services.

Il importe ensuite le module ADDSDeployment et utilise la commande Install-ADDSForest pour configurer le service AD.

Les options spécifiées incluent la spécification du nom de domaine et du nom Netbios, ainsi que l'emplacement des bases de données et des journaux.

Pour pouvoir accéder à distance à l'Active Directory et pour que le site puisse correctement créer des utilisateurs et changer leur mot de passe, il faut installer un serveur SSH :

```
# Installer le serveur OpenSSH
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

# Démarrer le service sshd
Start-Service -Name "sshd"

# Régler en mode automatique pour le démarrage du ssh
Set-Service -Name "sshd" -StartupType Automatic

# Vérifier les paramètres
Get-Service -Name "sshd" | Select-Object *

# Autoriser le port 22/TCP dans le Pare-feu Windows
New-NetFirewallRule -Name "SSH" `
-DisplayName "SSH" `
-Description "Allow SSH" `
-Profile Any `
-Direction Inbound `
-Action Allow `
-Protocol TCP `
-Program Any `
-LocalAddress Any `
-RemoteAddress Any `
-LocalPort 22 `
-RemotePort Any
```

Pour désactiver la complexité des mots de passe pour que chaque utilisateur puissent créer leur mot de passe, il suffit d'exécuter la commande suivante :

```
PS C:\Users\Administrateur> Set-ADDefaultDomainPasswordPolicy -ComplexityEnabled $false -MinPasswordLength 0

applet de commande Set-ADDefaultDomainPasswordPolicy à la position 1 du pipeline de la commande
Fournissez des valeurs pour les paramètres suivants :
Identity: virtuallaunch.local
PS C:\Users\Administrateur>
```

Pour désactiver le changement de mot de passe obligatoire dans l'AD, il suffit d'exécuter la commande suivante :

```
Set-ADDefaultDomainPasswordPolicy -Identity (domaine).local -ComplexityEnabled $false -MaxPasswordAge 0 -MinPasswordAge 1 -PasswordHistoryCount 0 -MinPasswordLength 0
```

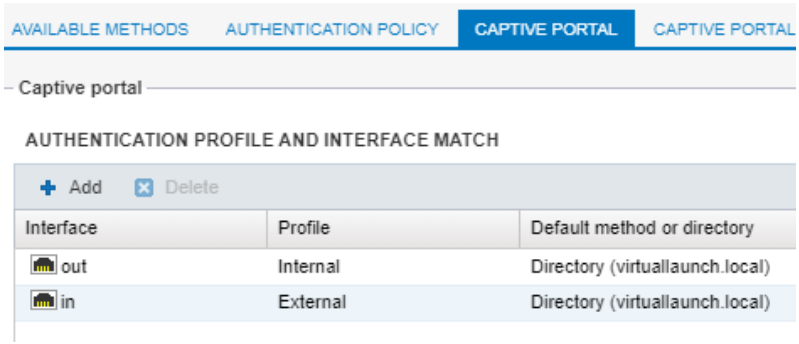
Note : Le site vérifie tout de même si le mot de passe fait au moins 8 caractères avant la modification de chaque mot de passe.

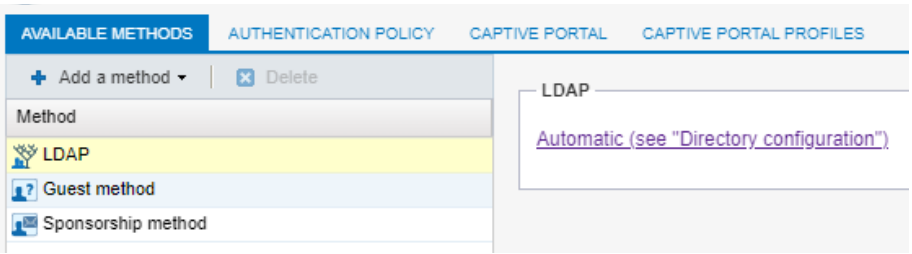
VPN SSL

CONFIGURATION D'UN TUNNEL

Prérequis :

- Un annuaire interne ou externe doit être configuré
- Un profil du portail captif doit être rattaché à l'interface depuis laquelle les utilisateurs se connectent
- Une méthode d'authentification doit être configurée
- Un PKI fournissant les certificats pour le serveur et les clients.





La première étape de mise en œuvre d'un tunnel VPN SSL est l'authentification de l'utilisateur via le portail captif, ce qui signifie que :

- Un annuaire externe ou interne doit être configuré au niveau du firewall.
- Un profil du portail captif doit être rattaché à l'interface depuis laquelle les utilisateurs se connectent.
- Une méthode d'authentification doit être configurée.

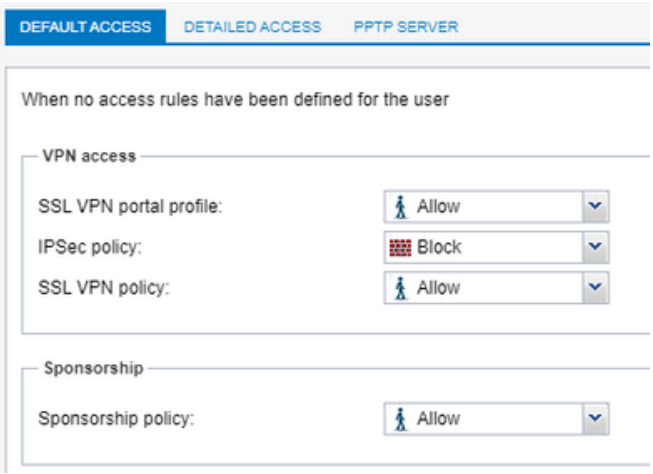
Les méthodes d'authentification possibles pour le service VPN SSL sont les méthodes explicites qui nécessitent un couple identifiant et mot de passe, en l'occurrence LDAP, Kerberos ou Radius.

L'authentification entre le client et le serveur VPN SSL s'effectue par certificat. Pour cela, une autorité de certification racine (CA) existe dans la configuration usine de tous les firewalls Stormshield Network. Cette CA est nommée sslvpn-full-default-authority, et elle contient un certificat serveur (qui identifie le serveur VPN SSL) et un certificat client.

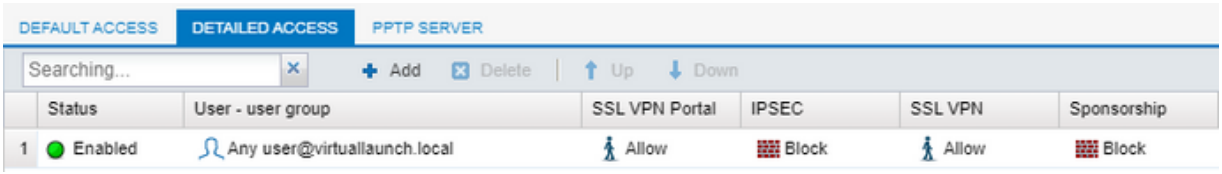
DROIT D'ACCÈS VPN SSL

Pour autoriser un utilisateur à monter un tunnel VPN SSL, vous devez lui attribuer les droits correspondants dans le menu "**Configuration/Utilisateur/Droits d'accès**".

Il est possible de choisir un accès par défaut indépendamment de l'utilisateur connecté dans l'onglet "**Accès détaillé/VPN SSL**". Sélectionnez "Autoriser" dans le champ "Politique VPN SSL" par défaut.

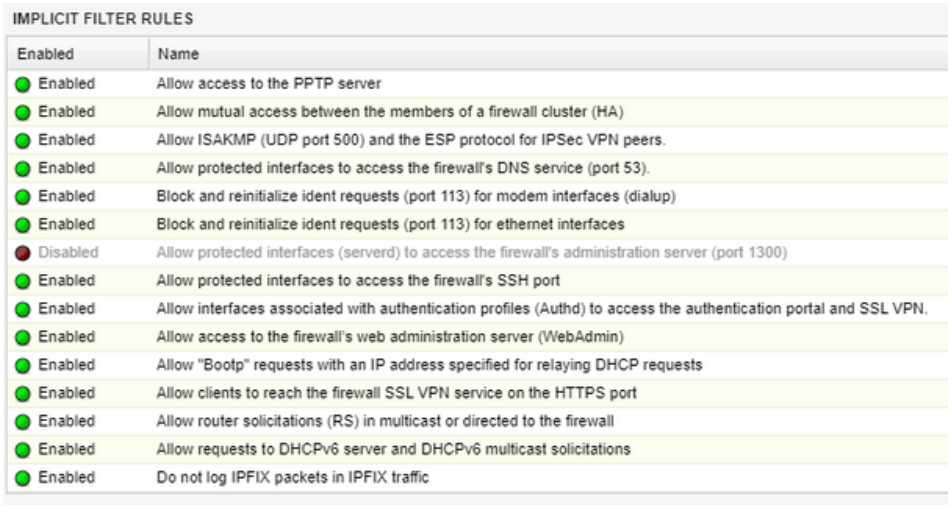


Cependant, une gestion plus fine des droits d'accès est préconisée en conservant la valeur de la politique VPN SSL par défaut « Interdire » et en ajoutant des utilisateurs ou des groupes d'utilisateurs dans l'onglet "**Accès détaillé/Ajouter**" avec les droits VPN SSL définis sur "**Autoriser**".



RÈGLE DE FILTRAGE IMPLICITE POUR LE VPN SSL

Pour permettre aux clients VPN SSL d'accéder au portail d'authentification sur les interfaces associées aux profils d'authentification du firewall, la règle de filtrage implicite nommée : "**Autoriser l'accès au portail d'authentification et au VPN SSL pour les interfaces associées aux profils d'authentification (Authd)**" doit être activée.

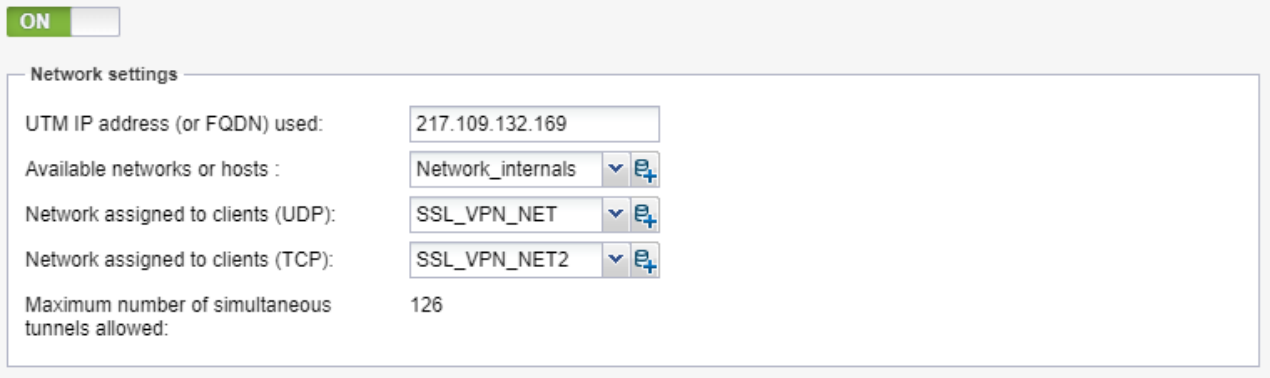


IMPLICIT FILTER RULES

Enabled	Name
Enabled	Allow access to the PPTP server
Enabled	Allow mutual access between the members of a firewall cluster (HA)
Enabled	Allow ISAKMP (UDP port 500) and the ESP protocol for IPSec VPN peers.
Enabled	Allow protected interfaces to access the firewall's DNS service (port 53).
Enabled	Block and reinitialize ident requests (port 113) for modem interfaces (dialup)
Enabled	Block and reinitialize ident requests (port 113) for ethernet interfaces
Disabled	Allow protected interfaces (serverd) to access the firewall's administration server (port 1300)
Enabled	Allow protected interfaces to access the firewall's SSH port
Enabled	Allow interfaces associated with authentication profiles (Authd) to access the authentication portal and SSL VPN.
Enabled	Allow access to the firewall's web administration server (WebAdmin)
Enabled	Allow "Bootp" requests with an IP address specified for relaying DHCP requests
Enabled	Allow clients to reach the firewall SSL VPN service on the HTTPS port
Enabled	Allow router solicitations (RS) in multicast or directed to the firewall
Enabled	Allow requests to DHCPv6 server and DHCPv6 multicast solicitations
Enabled	Do not log IPFIX packets in IPFIX traffic

CONFIGURATION DU SERVICE VPN SSL

Le service VPN SSL peut être configuré dans le menu Configuration VPN/VPN SSL.



Nous pouvons retrouver :

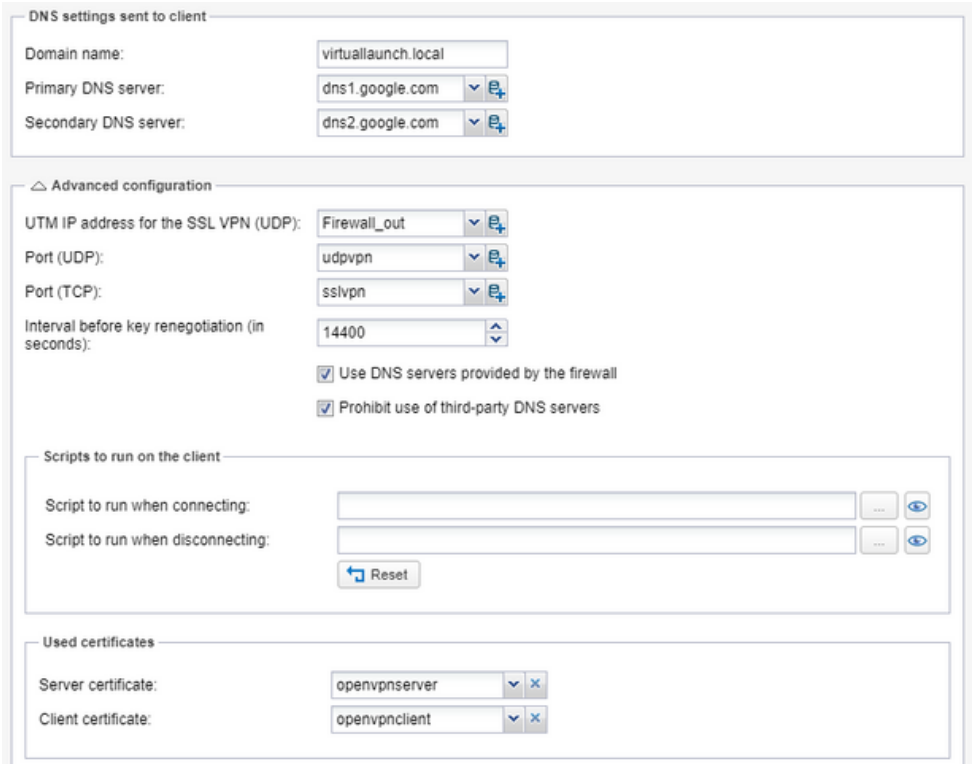
Adresse IP (ou FQDN) de l'UTM utilisée : il s'agit de l'adresse sur laquelle vont se connecter les clients VPN SSL (adresse publique la plupart du temps). Attention, la saisie d'un FQDN induit une résolution de noms via un service DNS.

Réseaux ou hôtes disponibles : machines ou réseaux auxquels les utilisateurs peuvent avoir accès une fois le tunnel établi (l'accès dépend néanmoins de la politique de filtrage active). Il est possible de choisir l'objet Any. Dans ce cas, tous les flux du client VPN passent par le tunnel et sont soumis aux opérations de filtrage et de NAT du firewall.

Réseau assigné aux clients (UDP) : réseau attribué aux clients nomades une fois le tunnel établi via le protocole UDP. La valeur minimale pouvant être choisie ici est un réseau de /29.

Réseau assigné aux clients (TCP) : réseau attribué aux clients nomades une fois le tunnel établi via le protocole TCP. La valeur minimale pouvant être choisie ici est un réseau de /29.

Nombre maximal de tunnels simultanés autorisés : paramètre non configurable dans la GUI. Il indique le nombre maximal de tunnels (clients) autorisés, c'est-à-dire le minimum entre le nombre de tunnels autorisés pour le modèle du firewall et le nombre de tunnels possibles calculé à partir du réseau assigné aux clients.



Ensuite, dans les paramètres DNS envoyés au client, nous retrouvons :

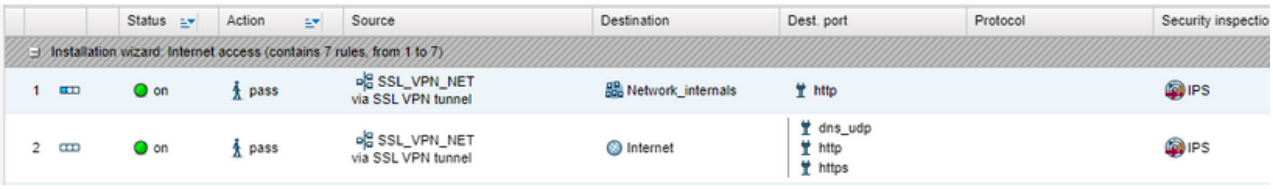
- Nom de domaine : il s'agit en général du domaine dont dépendent les réseaux accessibles par le client
- Serveur DNS primaire (et secondaire): interne, si le client doit pouvoir accéder à des ressources locales. Sinon, le choix d'un serveur public est autorisé.

Et nous avons la configuration avancée ou nous pouvons retrouver :

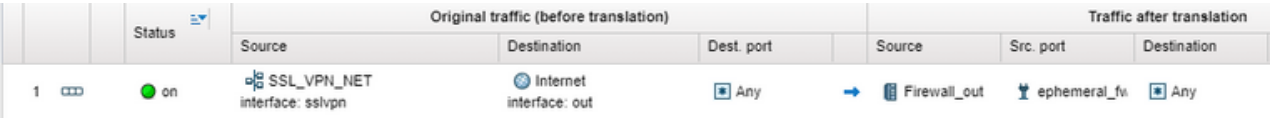
- Adresse IP de l'UTM pour le VPN SSL (UDP) : il s'agit de l'adresse à laquelle vont se connecter les clients du VPN SSL s'ils sont configurés pour utiliser l'UDP (adresse publique la plupart du temps).
- Port (UDP) : port d'écoute UDP du service VPN SSL.
- Port(TCP) : port d'écoute TCP du service VPN SSL.
- Utiliser les serveurs DNS fournis par le firewall lorsque cette option est choisie, le client VPN SSL ajoutera les serveurs DNS qui ont été récupérés via le tunnel VPN SSL à la configuration réseau du poste de travail du client.
- Interdire l'utilisation des serveurs DNS tiers : lorsque cette option est choisie, le poste de travail du client utilisera uniquement les serveurs DNS qui ont été récupérés via le tunnel VPN SSL.

FILTRAGE ET NAT

Il est nécessaire de définir des règles de filtrage explicites pour la gestion du trafic provenant des tunnels



Une translation d'adresses peut être mise en œuvre si des clients doivent utiliser le VPN SSL pour accéder à Internet.



La règle de filtrage n°1 permet l'initiation de connexions à partir des clients VPN SSL et à destination de tous les réseaux internes (Network Internals),
La règle de filtrage n°2 permet l'initiation de connexions à partir des clients VPN SSL et à destination d'internet ; dans ce cas, une règle de NAT doit également être ajoutée.

La configuration est terminée, il suffira de vérifier si le VPN est fonctionnel en téléchargeant le client VPN sur le site : https://mystormshield.eu

Une fois le client téléchargé, il vous restera à ouvrir le client VPN et à renseigner les trois paramètres :

- L'adresse IP ou le FQDN du firewall à contacter
- L'identifiant de l'utilisateur disposant des droits pour le VPN SSL
- Le mot de passe associé à l'utilisateur

Une fois le tunnel établi, le poste client dispose d'une interface spécifique pour le tunnel VPN SSL, dont l'adresse IP fait partie de l'objet "Réseau" assigné au client dans la configuration serveur.