

Sprint 2

Sprint réalisé par Vincent LAGOGUÉ et Thomas PEUGNET.

L'objectif de ce sprint est de:

- Configurer des accès SSH à des utilisateurs à partir d'une base LDAP
- Configurer des accès à un site web en HTTP et HTTPS à partir d'une base LDAP
- Configurer des accès pour une connexion à un serveur OpenVPN à partir d'une base LDAP

Une conclusion et un résumé de ce qui a été fait avec succès et de ce qui a échoué se trouve à la fin de ce rapport, en dernière page.

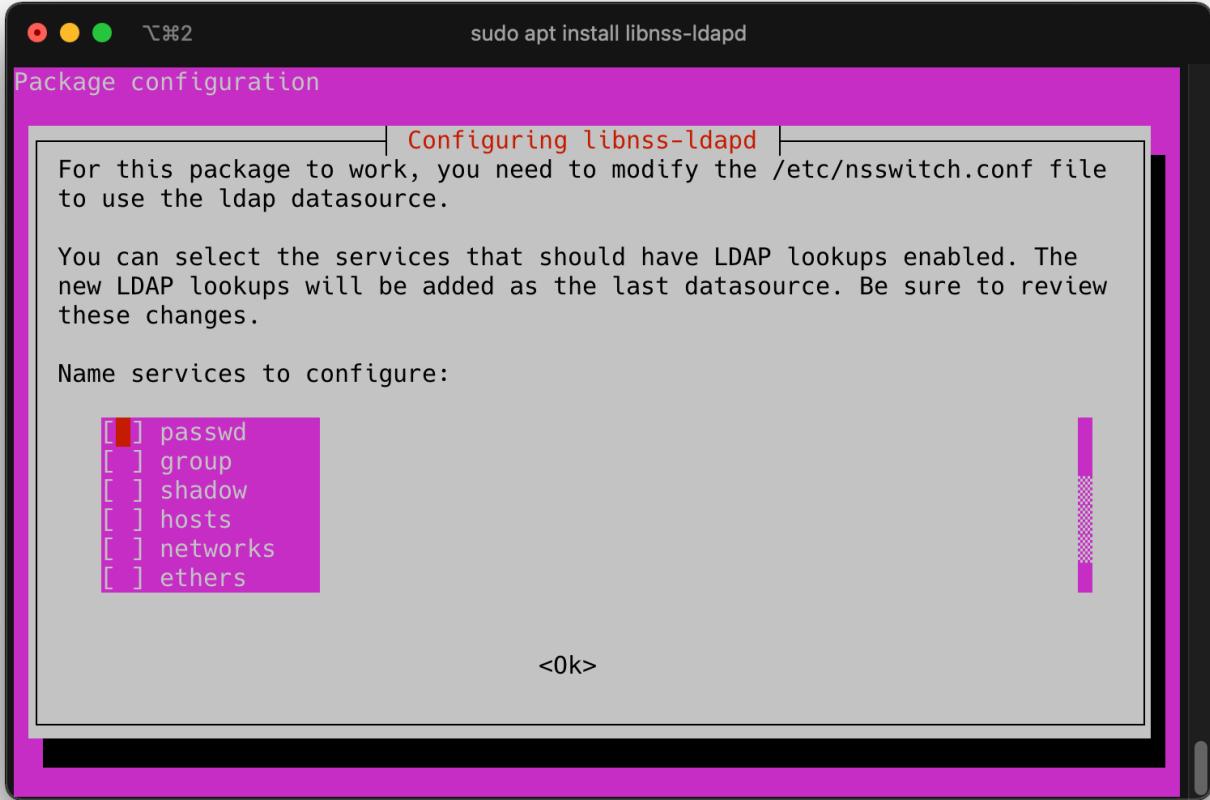
Configuration SSH x LDAP

Configuration de PAM

```
$ sudo apt install libnss-ldap
```

Par la suite, un écran de configuration va être affiché, il s'agit de remplir les informations suivantes :

- URI du serveur LDAP : Efrei.fr:636
- Base de recherche : dc=Efrei,dc=fr
- Make local root Database admin : Yes
- Does database require login : No
- LDAP account for root : cn=admin,dc=Efrei,dc=fr
- Configuration des services à configurer : Aucun



Utiliser la commande suivante pour vérifier le bon fonctionnement :

```
$ LDAPTLS_REQCERT=never ldapsearch -H ldaps://Efrei.fr:636 -b 'dc=Efrei,dc=fr' -x  
uid=thomas.peugnet -LLL
```

On obtient le résultat suivant :

```
root@tpnagios-1:~  
  
@UBUNTU ~ 13:07:54  
$ LDAPTLS_REQCERT=never ldapsearch -H ldaps://Efrei.fr:636 -b 'dc=Efrei,dc=f  
r' -x uid=thomas.peugnet -LLL  
dn: uid=thomas.peugnet,ou=users,dc=Efrei,dc=fr  
objectClass: person  
objectClass: top  
objectClass: organizationalPerson  
objectClass: inetOrgPerson  
objectClass: posixAccount  
uidNumber: 6001  
gidNumber: 6001  
homeDirectory: /home/thomas  
loginShell: /bin/bash  
uid: thomas.peugnet  
sn: peugnet  
cn: thomas peugnet  
mail: thomas.peugnet@efrei.fr  
  
@UBUNTU ~ 13:07:59  
$
```

On modifie les fichiers `nsswitch.conf` et `nsLCD.conf` pour avoir le résultat suivant:

`nsswitch.conf` (attention aux lignes `passwd`, `group`, `shadow` et `gshadow`):

```
# /etc/nsswitch.conf  
#  
# Example configuration of GNU Name Service Switch functionality.  
# If you have the `glibc-doc-reference` and `info` packages installed, try:  
# `info libc "Name Service Switch"' for information about this file.  
  
passwd:      ldap files systemd  
group:       ldap files systemd  
shadow:      ldap files  
gshadow:     files  
  
hosts:       files dns  
networks:    files  
  
protocols:   db files  
services:    db files  
ethers:      db files  
rpc:         db files  
  
netgroup:    nis
```

`nsLCD.conf`:

```

# /etc/nslcd.conf
# nslcd configuration file. See nslcd.conf(5)
# for details.

# The user and group nslcd should run as.
uid nslcd
gid nslcd

# The location at which the LDAP server(s) should be reachable.
uri ldaps://Efrei.fr:636

# The search base that will be used for all queries.
base dc=Efrei,dc=fr

# The LDAP protocol version to use.
#ldap_version 3

# The DN to bind with for normal lookups.
#binddn cn=annonymous,dc=example,dc=net
#bindpw secret

# The DN used for password modifications by root.
#rootpwmoddn cn=admin,dc=example,dc=com

# SSL options
#ssl off
tls_reqcert never
tls_cacertfile /etc/ldap/ssl/cert.pem

# The search scope.
#scope sub

```

On redémarre le service :

```
$ sudo service nslcd restart
```

On vérifie que le service a bien redémarré avec la commande suivante :

```
$ getent passwd | grep thomas.peugnet
```

```
root@tpnagios-1:~  
@UBUNTU ~ 13:22:24  
$ getent passwd | grep thomas.peugnet  
thomas.peugnet:*:6001:6001:thomas peugnet:/home/thomas:/bin/bash  
@UBUNTU ~ 13:22:39  
$
```

Note: Une erreur avait été faite lors de la création de l'utilisateur ci-dessus, avec les mauvais UIDs.

Si plusieurs utilisateurs ont les mêmes UIDs, il est possible de les changer via l'url de LAM:

<http://192.168.1.28/lam/templates/login.php>

Note: Si, en se connectant avec un utilisateur, on se retrouve connecté sur le compte d'un autre utilisateur, il est possible qu'il existe un conflit sur les UIDs.

On crée les dossiers personnels des utilisateurs avec les bonnes permissions à l'aide des commandes suivantes:

```
$ mkdir /home/tom && chown tom.thioulouse:teachers -R /home/tom  
$ mkdir /home/thomas && chown thomas.peugnet:students -R /home/thomas  
$ mkdir /home/alexis && chown alexis.plessias:students -R /home/alexis
```

Enfin, on essaye de se connecter via une autre instance :

```
$ ssh alexis.plessias@192.168.1.28
```

Le mot de passe étant défini dans le fichier `.ldif` du TP01.

On obtient le résultat suivant :

```
● ● ● 7:33 ssh alexis.plessias@192.168.1.28
↳ thomas@Thomass-MacBook-Pro ~
    ssh alexis.plessias@192.168.1.28
    alexis.plessias@192.168.1.28's password: 130 ↵
    Welcome to Ubuntu 22.10 (GNU/Linux 5.15.131-2-pve x86_64)

    * Documentation: https://help.ubuntu.com
    * Management: https://landscape.canonical.com
    * Support: https://ubuntu.com/advantage
Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

alexis.plessias@Efrei:~$
```

Gestion des accès SSH

On modifie le fichier de configuration SSH, pour n'autoriser que les membres du groupe `teachers` à se connecter via SSH, en ajoutant à la fin du fichier :

```
AllowGroups teachers
```

On modifie également le fichier de configuration `/etc/pam.d/sshd` en ajoutant la ligne suivante :

```
auth required pam_group.so
```

Ce qui donne le résultat suivant :

```
vim /etc/pam.d/sshd

42 # Read environment variables from /etc/environment and
43 # /etc/security/pam_env.conf.
44 session required pam_env.so # [1]
45 # In Debian 4.0 (etch), locale-related environment variables were moved
46 # to
47 session required pam_env.so user_readenv=1 envfile=/etc/default/l
ocale
48
49 # SELinux needs to intervene at login time to ensure that the process st
arts
50 # in the proper default security context. Only sessions which are inten
ded
51 # to run in the user's context should be run after this.
52 session [success=ok ignore=ignore module_unknown=ignore default=bad]
      pam_selinux.so open
53
54 # Standard Un*x password updating.
55 @include common-password
56
57 auth required pam_group.so
~
~
~
~

/etc/pam.d/sshd
```

57,1

Bot

On redémarre le service ssh avec la commande suivante :

```
$ systemctl restart ssh.service
```

Puis, on re-tente de se connecter en ssh avec un utilisateur du groupe `students` :

```
$ ssh alexis.plessias@192.168.1.28
```

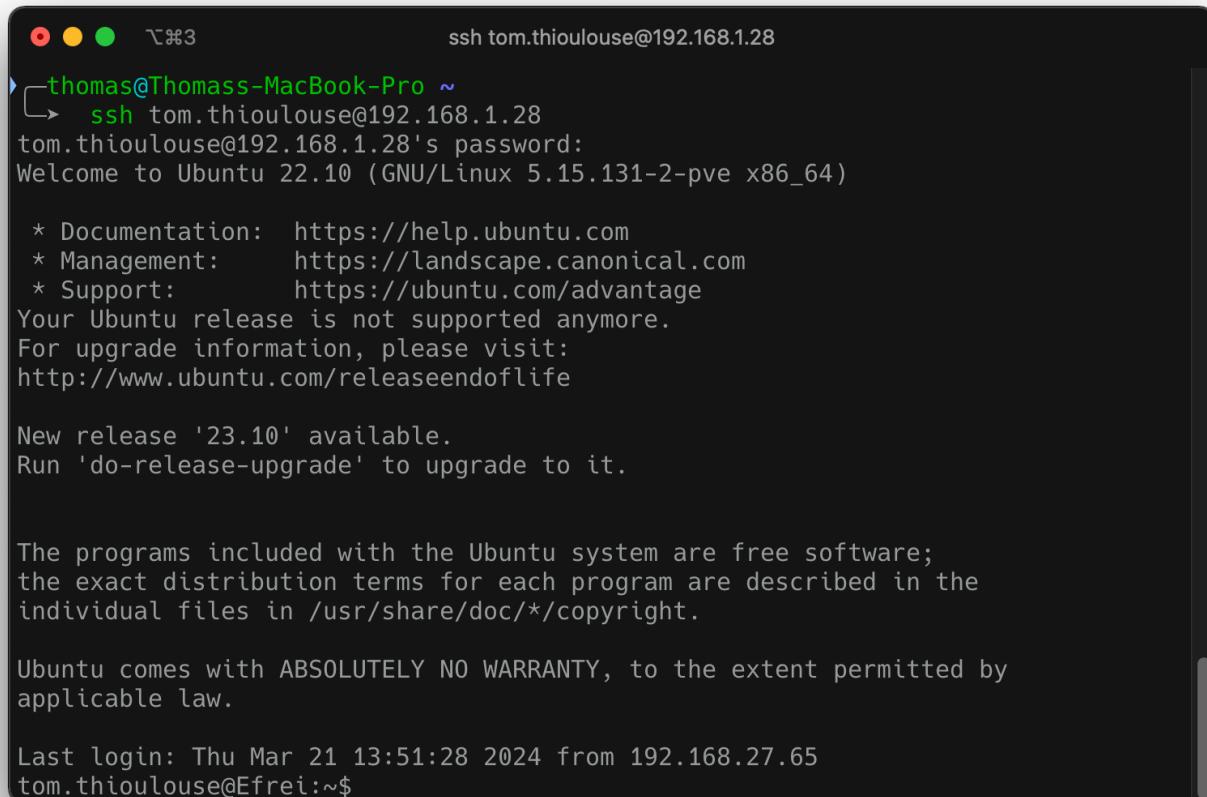
On obtient le résultat suivant :

```
ssh alexis.plessias@192.168.1.28

Connection to 192.168.1.28 closed.
↳ thomas@Thomass-MacBook-Pro ~
    ↳ ssh alexis.plessias@192.168.1.28
alexis.plessias@192.168.1.28's password:
Permission denied, please try again.
alexis.plessias@192.168.1.28's password:
```

Si on tente de se connecter avec un utilisateur du groupe `teachers` :

```
$ ssh tom.thioulouse@192.168.1.28
```



A screenshot of a macOS terminal window titled "ssh tom.thioulouse@192.168.1.28". The session starts with a password prompt for "tom.thioulouse@192.168.1.28". It then displays the standard Ubuntu 22.10 welcome message, including links for documentation, management, and support. It also mentions that the release is no longer supported and provides upgrade information. A note about a new release '23.10' is present, along with instructions to run 'do-release-upgrade'. The terminal then shows the standard copyright notice and a warning about no warranty. Finally, it displays the last login information: "Last login: Thu Mar 21 13:51:28 2024 from 192.168.27.65" followed by the prompt "tom.thioulouse@Efrei:~\$".

Note: Si l'erreur `Could not chdir to home directory /home/X: No such file or directory`, c'est que le dossier de l'utilisateur n'existe pas. Si une erreur de `Permission Denied` survient, c'est le `chown` qui n'a pas été correctement effectué.

Configuration Apache x LDAP

Configuration de Apache2

Installation du service Apache2 et activation du module d'authentification :

```
# Installation
$ apt install apache2

# Activation du module
$ sudo a2enmod authnz_ldap

# Restart du service
$ systemctl restart apache2
```

Modification du VirtualHost par défaut d'Apache pour lui ajouter une Basic Auth:

Le fichier `/etc/apache2/sites-available/000-default.conf` doit donc avoir le contenu suivant:

```
<AuthnProviderAlias ldap myldap>
    AuthLDAPURL "ldap://Efrei.fr/ou=users,dc=Efrei,dc=fr"
</AuthnProviderAlias>

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the LogLevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf

    <Directory "/var/www/html">
        AuthType Basic
        AuthName "Top Secret"
        AuthBasicProvider myldap
        Require valid-user
        LogLevel trace1
    </Directory>
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

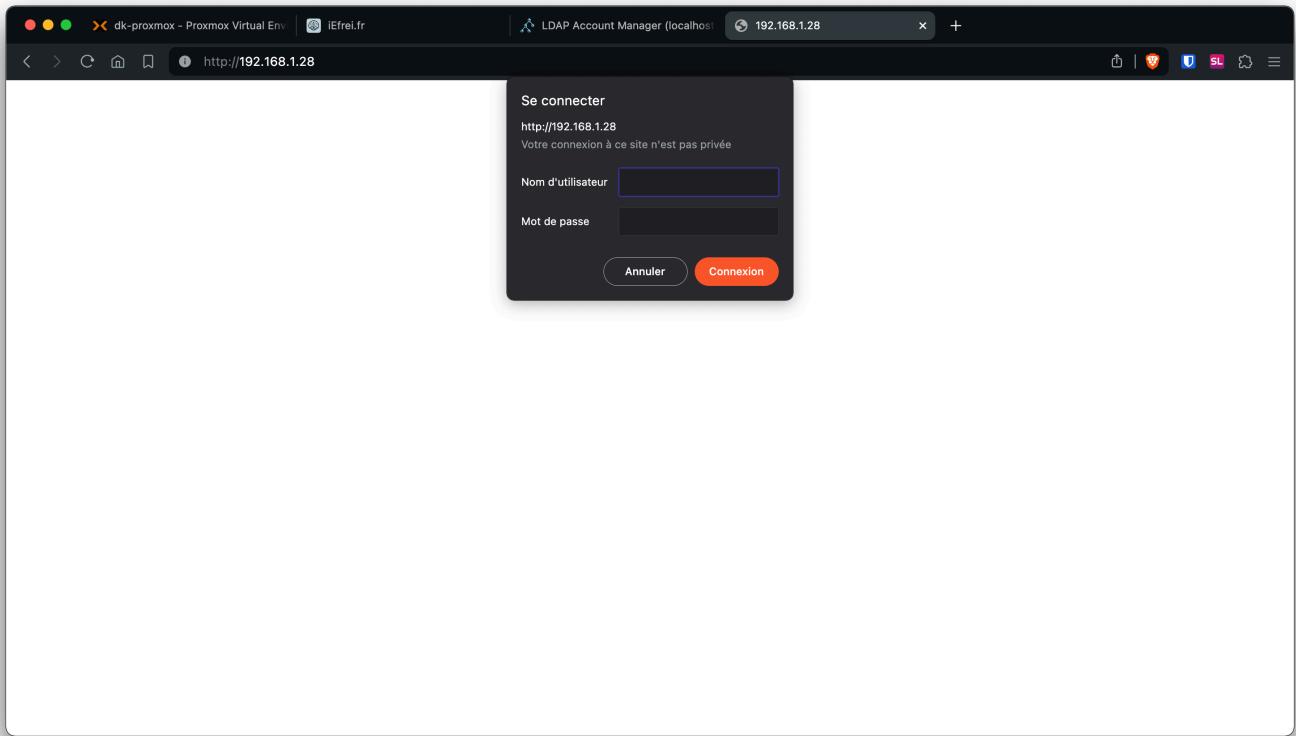
On redémarre le service:

```
$ systemctl restart apache2
```

On modifie le contenu de la page sur laquelle le VHost redirige avec le contenu suivant:

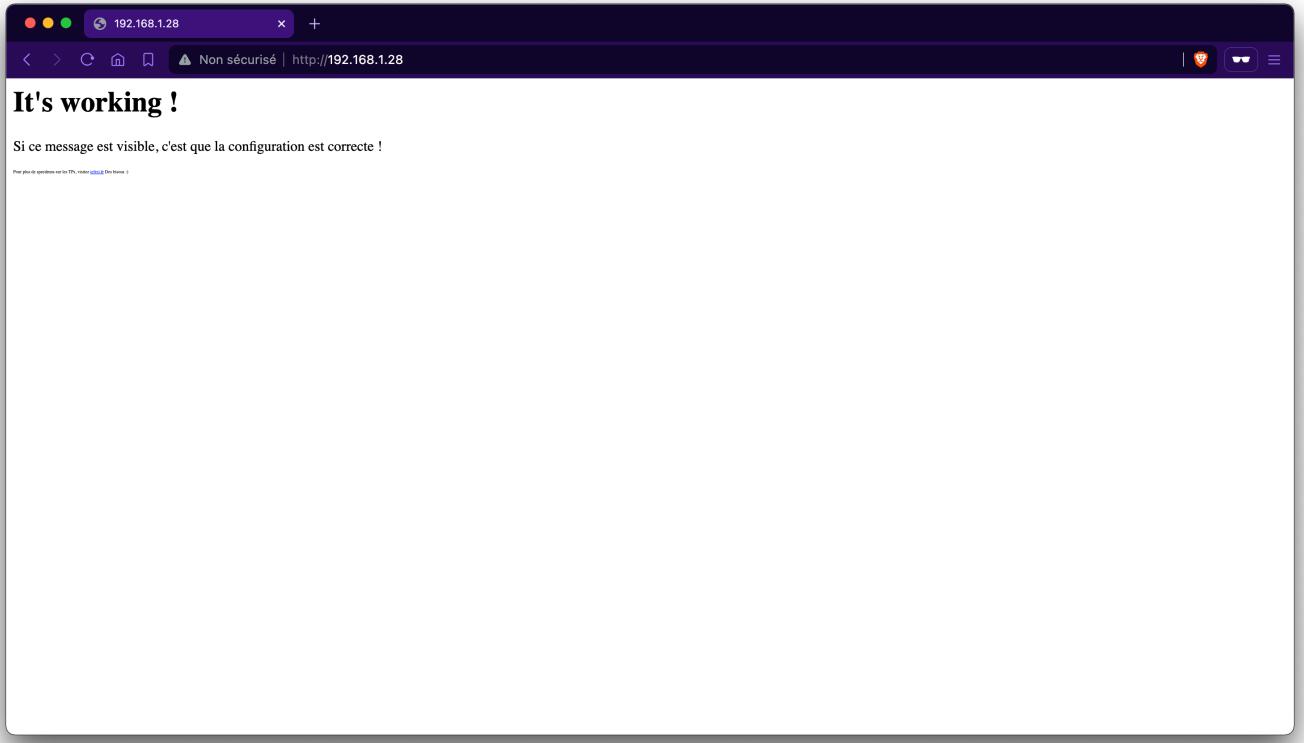
```
<h1>It's working !</h1>
<p>Si ce message est visible, c'est que la configuration est correcte !</p>
<p style="font-size:5px">Meilleur site de speedruns sur les TPs, visitez <a href="https://iefrei.fr">iefrei.fr</a> !</p>
```

On accède à la page sur l'URL <http://192.168.1.28>, et on constate le résultat suivant :



Cela nous indique que la configuration de la Basic Auth fonctionne correctement.

Si on entre les informations de l'utilisateur `alexis.plessias` :



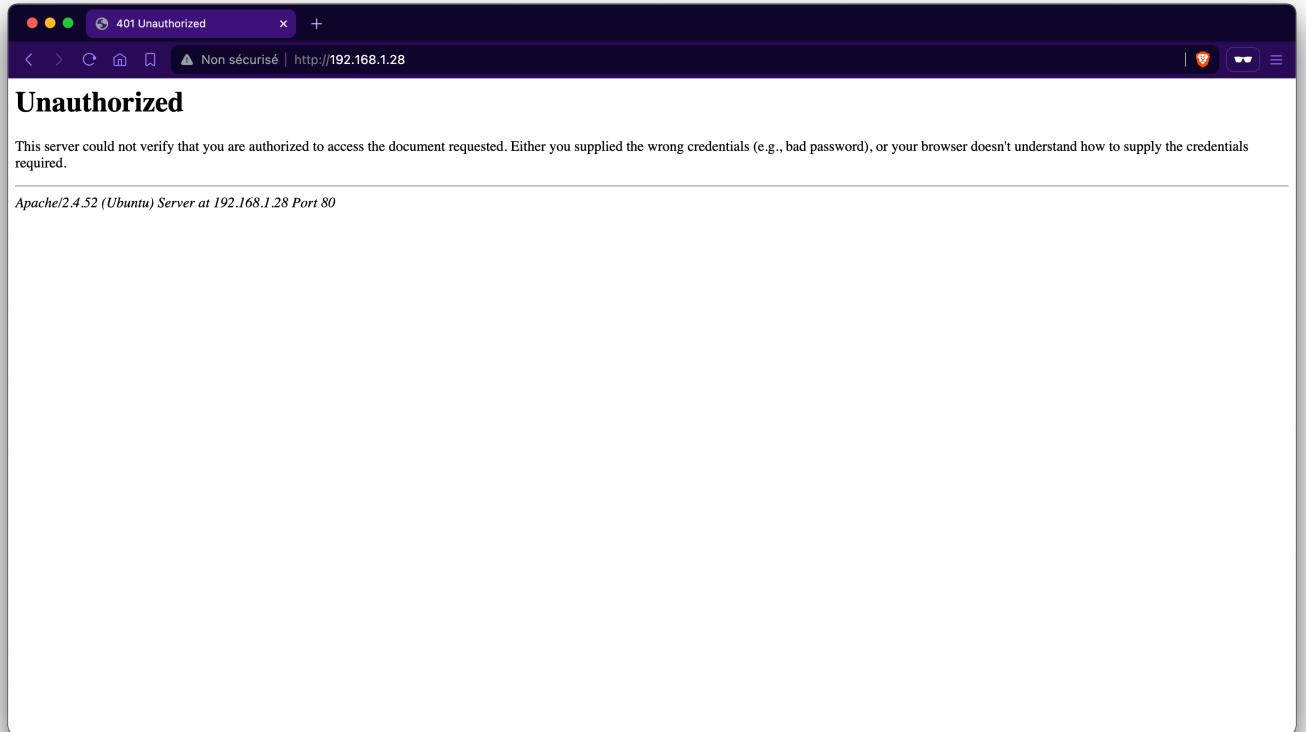
Restriction des accès au groupe teachers

Modification de la ligne 2 de la configuration du VHost avec le contenu suivant :

```
<AuthnProviderAlias ldap myldap>
    AuthLDAPURL "ldap://Efrei.fr/ou=users,dc=Efrei,dc=fr?uid?sub?(gidNumber=6001)"
</AuthnProviderAlias>
```

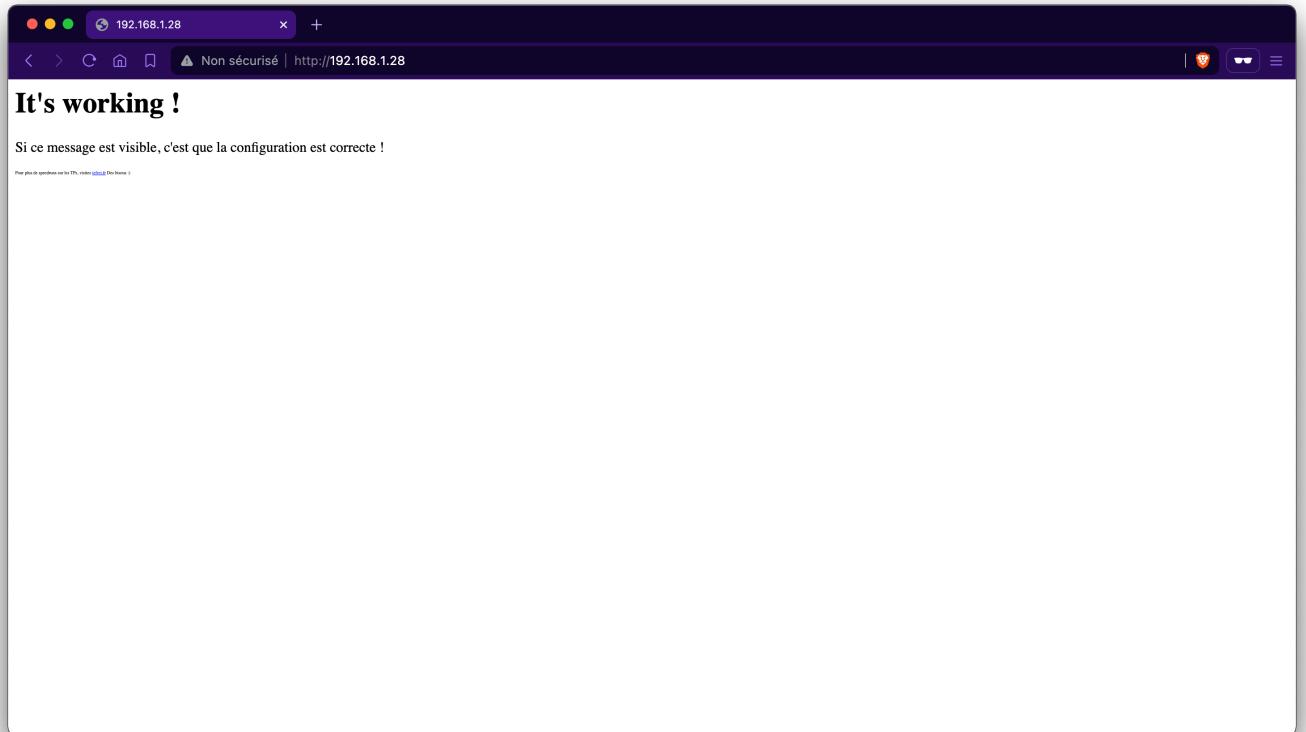
Avec 6001 le GID du groupe teachers .

Avec la connexion en tant qu'utilisateur alexis.plessias du groupe students :



Note: L'écran ci-dessus arrive lorsque l'on clique sur le bouton `Annuler`, après plusieurs tentatives infructueuses de connexion.

Avec l'utilisateur `tom` qui est membre du groupe `teachers` :



Amélioration - HTTPS

Création d'un certificat pour l'HTTPS:

```
$ openssl genkey -algorithm RSA -out /etc/ssl/private/https_key.key -pkeyopt  
rsa_keygen_bits:2048  
$ openssl req -new -key /etc/ssl/private/https_key.key -out https_request.csr  
$ openssl x509 -signkey /etc/ssl/private/https_key.key -in https_request.csr -req -days 365 -  
out /etc/ssl/certs/https_cert.crt
```

Pour activer l'HTTPS sur notre serveur web, nous allons commencer par activer le module `ssl` à l'aide de la commande suivante:

```
$ a2enmod ssl
```

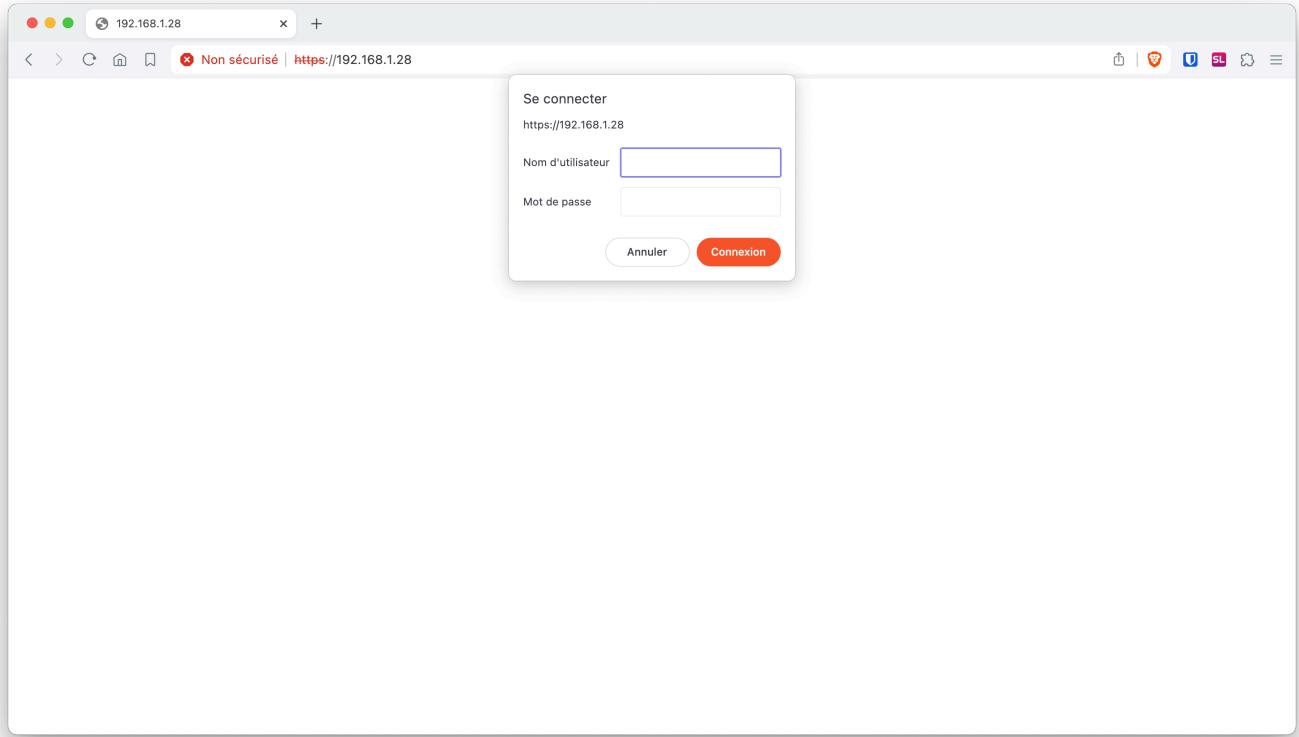
On effectue une backup du fichier de configuration original.

```
$ cd /etc/apache2/sites-available  
$ cp default-ssl.conf default-ssl.conf.old  
$ echo "" > default-ssl.conf
```

On modifie notre configuration Apache dans notre fichier `default-ssl.conf` par la configuration suivante:

```
<VirtualHost *:443>  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/html  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
  
    SSLEngine on  
    SSLCertificateFile /etc/ssl/certs/https_cert.crt  
    SSLCertificateKeyFile /etc/ssl/private/https_key.key  
  
    <Directory "/var/www/html">  
        AuthType Basic  
        AuthName "Top Secret"  
        AuthBasicProvider myldap  
        Require valid-user  
        LogLevel trace1  
    </Directory>  
</VirtualHost>
```

En nous rendant sur le site en `https` on obtient le résultat suivant:



Nous pouvons constater que cela fonctionne correctement.

Configuration OpenVPN x LDAP

Note: Cette partie n'aboutit pas à une configuration OpenVPN fonctionnelle. Elle détaille l'installation de 4 façons différentes, dont l'une avec succès, la configuration qui, normalement devrait fonctionner, mais la finalité demeure être une impossibilité de lancer le serveur OpenVPN, et donc d'initier une connexion depuis un client distant.

Installation de OpenVPN

```
$ apt install openvpn
```

Vérifier la bonne installation en se rendant dans le dossier suivant :

```
$ cd /usr/share/doc/openvpn/examples
```

```
● ● ●  ~%2          root@Efrei:/usr/share/doc/openvpn/examples  
@UBUNTU /usr/share/doc/openvpn/examples  14:28:44  
$ cd /usr/share/doc/openvpn/examples  
@UBUNTU /usr/share/doc/openvpn/examples  14:29:35  
$ ls  
sample-config-files  sample-keys  sample-scripts  
@UBUNTU /usr/share/doc/openvpn/examples  14:29:47  
$ █
```

On récupère l'adresse IP de notre instance par la commande `hostname -I`.

On modifie le contenu de la configuration du fichier `sample-config-files/client.conf` avec la ligne suivante (variable selon votre IP):

```
remote 192.168.1.28 1194
```

```
● ● ●  ~%2          vim sample-config-files/client.conf  
33 # Are we connecting to a TCP or  
34 # UDP server? Use the same setting as  
35 # on the server.  
36 ;proto tcp  
37 proto udp  
38  
39 # The hostname/IP and port of the server.  
40 # You can have multiple remote entries  
41 # to load balance between the servers.  
42 #remote my-server-1 1194  
43 #;remote my-server-2 1194  
44 remote 192.168.1.28 1194  
45 █  
46 # Choose a random host from the remote  
47 # list for load-balancing. Otherwise  
48 # try hosts in the order specified.  
49 ;remote-random  
50  
51 # Keep trying indefinitely to resolve the  
52 # host name of the OpenVPN server. Very useful  
53 # on machines which are not permanently connected  
54 # to the internet such as laptops.  
sample-config-files/client.conf      45,0-1      29%
```

On teste le bon fonctionnement de ce VPN à l'aide de la commande suivante :

```
# Se placer dans le dossier contenant les certificats
$ cd /usr/share/doc/openvpn/examples/sample-keys

# Lancer la configuration par défaut
$ openvpn --config ./sample-config-files/server.conf
```

Dû à des problèmes liés au kernel proxmox (`/dev/net/tun` inexistant dans cette version), nous passerons par l'utilisation d'un Docker en lieu et place de l'installer en bare metal.

On commence donc par créer notre environnement de travail. Notes issues du site [Docker officiel](#).

```
$ cd
$ mkdir openvpn && mkdir openvpn.ovpn-data-example && cd openvpn
$ OVPN_DATA="ovpn-data-example"
# Initialize the $OVPN_DATA container that will hold the configuration files and certificates.
The container will
# prompt for a passphrase to protect the private key used by the newly generated certificate
authority.
$ docker volume create --name $OVPN_DATA
$ docker run -v $OVPN_DATA:/etc/openvpn --rm kylemanna/openvpn ovpn_genconfig -u
udp://VPN.SERVERNAME.COM
$ docker run -v $OVPN_DATA:/etc/openvpn --rm -it kylemanna/openvpn ovpn_initpki
```

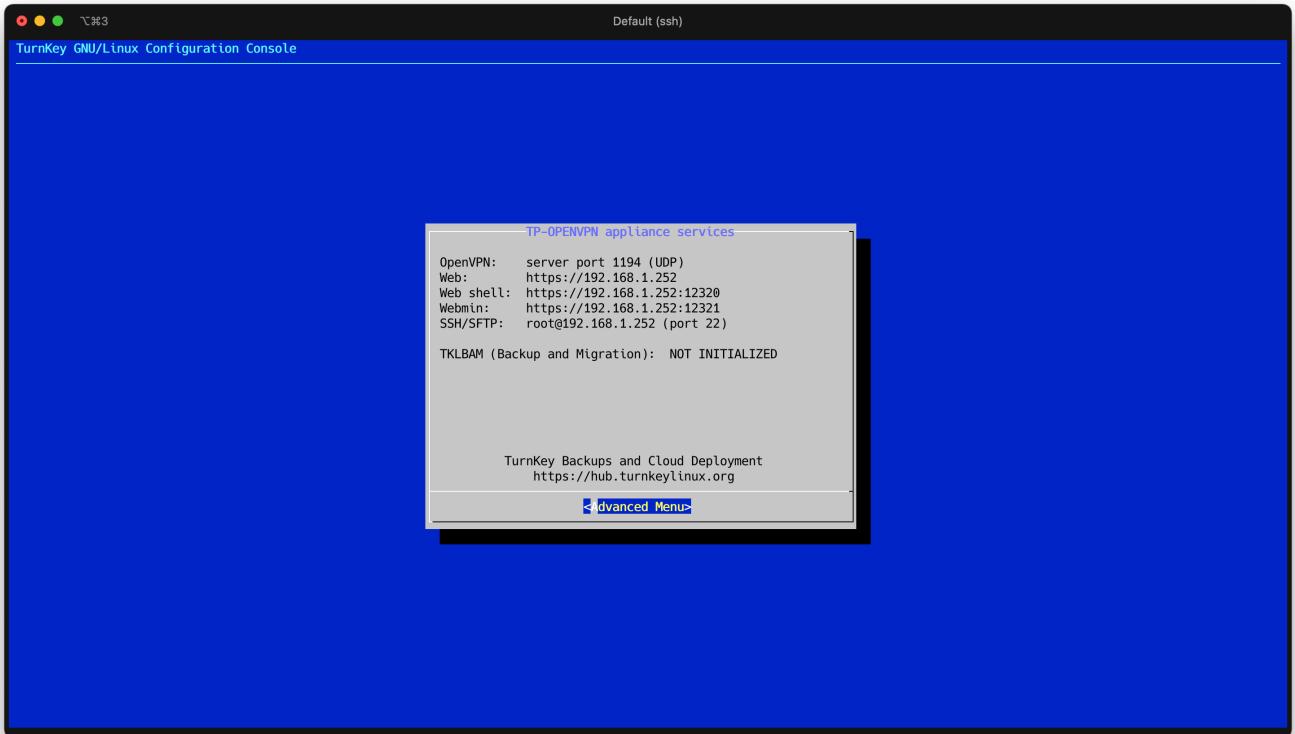
On start le conteneur :

```
$ docker run -v $OVPN_DATA:/etc/openvpn -p 1194:1194/udp --cap-add=NET_ADMIN kylemanna/openvpn
```

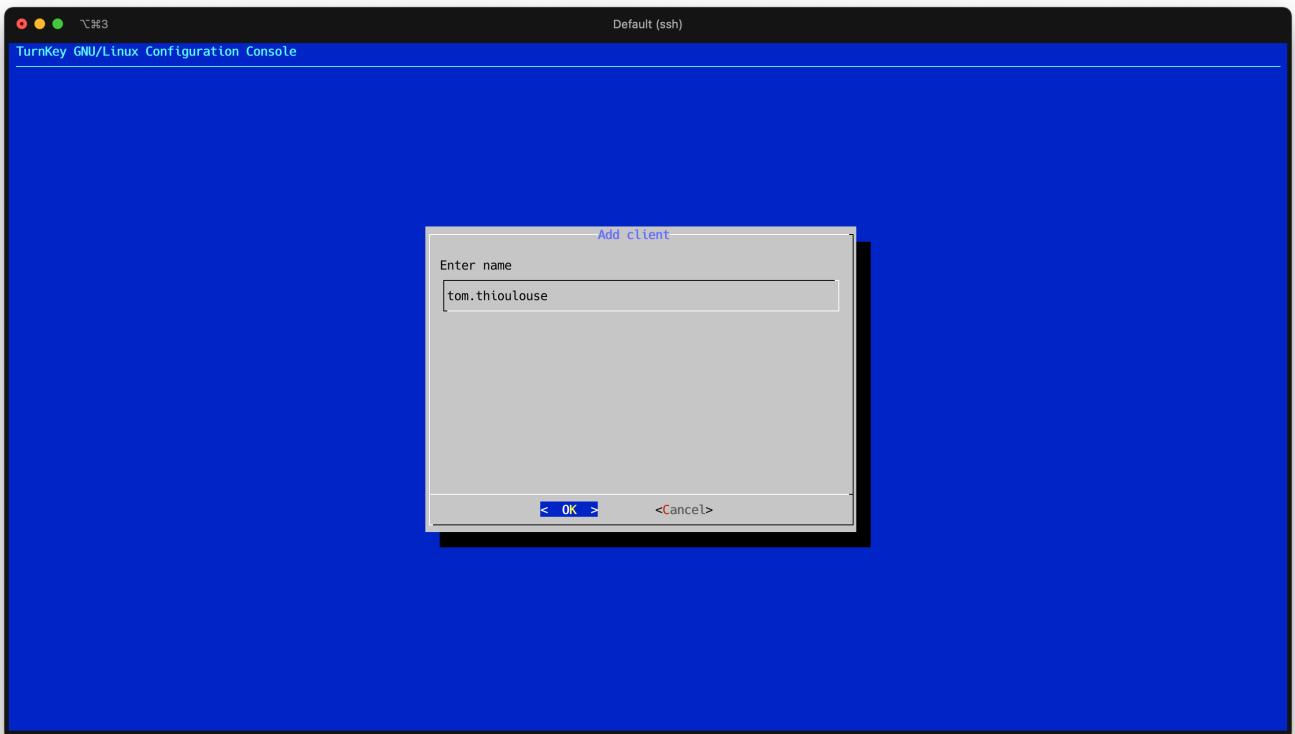
Mais finalement, on obtient le même problème de `/dev/net/tun` inexistant.

Nous essayons ensuite de passer par un conteneur template `lxc`, qui est une fonctionnalité Proxmox permettant de créer des conteneurs LXC directement à partir d'un template, dans notre cas OpenVPN.

Après avoir lancé le conteneur, nous obtenons à nouveau une erreur sur le `/dev/net/tun`, mais l'installation semble se poursuivre. A l'issue de cette dernière, nous obtenons le résultat suivant:



Nous ajoutons ensuite un nouveau client:



Une fois ceci fait, nous obtenons une URL nous permettant de télécharger ou voir la configuration du client `tom.thioulouse` à partir d'un lien de la forme

`http://192.168.1.252/profiles/1145ab468e54137e1f56079040561ab3296c3767/tom.thioulouse.ovpn`.

Nous obtenons le fichier de configuration du client. Après avoir essayé de se connecter, nous pouvons constater que cela fonctionne parfaitement. Il reste maintenant à intégrer l'utilisation du module LDAP pour l'authentification.

Configuration du plugin ldap

Nous commençons par créer notre configuration LDAP qui sera utilisée par le plugin pour l'authentification. Nous créons donc le fichier `/etc/openvpn/auth-ldap.conf` avec le contenu suivant:

```
<LDAP>
# Connexion au serveur LDAP
URL ldaps://192.168.1.28:636

# Paramètres de l'annuaire LDAP
BindDN cn=admin,dc=Efrei,dc=fr
Password bind_password
Timeout 15
TLSEnable no
FollowReferrals no

# Base de recherche pour les utilisateurs
BaseDN "ou=users,dc=Efrei,dc=fr"
SearchFilter "(uid=%u)"

#RequireGroup yes
#BaseDN "ou=groups,dc=Efrei,dc=fr"
#GroupSearchFilter "(memberUid=%u)"
#GroupName cn=vpn_users,ou=groups,dc=Efrei,dc=fr
</LDAP>
```

Nous ajoutons maintenant cette ligne dans `/etc/openvpn/server.conf` pour intégrer l'utilisation du plugin `openvpn-auth-ldap`.

```
plugin /usr/lib/openvpn/plugins/openvpn-plugin-auth-ldap.so "/etc/openvpn/auth-ldap.conf"
```

Nous avons donc une configuration qui a le contenu suivant:

```
root@tp-openvpn: /etc/openvpn
PUBLIC_ADDRESS: vpn.Efrei.fr (used by openvpn-addclient)

port 1194
proto udp
dev tun

keepalive 10 120

persist-key
persist-tun
user nobody
group nogroup

chroot /etc/openvpn/easy-rsa/keys/crl.jail
crl-verify /etc/openvpn/crl.pem

ca /etc/openvpn/easy-rsa/keys/ca.crt
dh /etc/openvpn/easy-rsa/keys/dh.pem
tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0
key /etc/openvpn/easy-rsa/keys/private/server.key
cert /etc/openvpn/easy-rsa/keys/issued/server.crt

ifconfig-pool-persist /var/lib/openvpn/server.ipp
client-config-dir /etc/openvpn/server.ccd
status /var/log/openvpn/server.log
verb 4

# virtual subnet unique for openvpn to draw client addresses from
# the server will be configured with x.x.x.1
# important: must not be used on your network
server 10.232.14.0 255.255.255.0
# push routes to clients to allow them to reach private subnets
push "route 10.0.1.0 255.255.255.0"

plugin /usr/lib/openvpn/plugins/openvpn-plugin-auth-ldap.so "/etc/openvpn/auth-ldap.conf"
```

1,1

Top

Enfin, nous redémarrons notre service à l'aide de la commande suivante:

```
$ systemctl restart openvpn@server
```

Hélas, encore une fois, le plugin `openvpn-plugin-auth-ldap.so` semble ne pas avoir été installé correctement, ou se trouve à un autre endroit comme en témoigne le `journalctl`.

Nous essayons donc, en dernière tentative, de cloner directement le code source du plugin et de l'installer manuellement.

```
$ git clone https://github.com/snowrider311/openvpn-auth-ldap
$ cd openvpn-auth-ldap/
$ ./ubuntu_16.04_lts_build.sh
```

Avec cette commande, nous obtenons ENFIN notre plugin d'authentification:

```
root@tp-openvpn: /usr/local/lib
root@tp-openvpn .../local/lib# ls
openvpn-auth-ldap.so  python2.7  python3.11
root@tp-openvpn .../local/lib#
```

Nous procémons à une modification du nom dans notre configuration de `server.conf` :

```
plugin /usr/local/lib/openvpn-auth-ldap.so "/etc/openvpn/auth-ldap.conf"
```

Nous n'obtenons cette fois-ci plus d'erreur lors du restart de notre service OpenVPN !

```
root@tp-openvpn: /etc/openvpn
~
~
~
~

root@tp-openvpn /etc/openvpn# systemctl status openvpn@server
* openvpn@server.service - OpenVPN connection to server
  Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled; preset: enabled)
  Active: activating (auto-restart) (Result: exit-code) since Thu 2024-04-04 07:43:33 UTC; 942ms ago
    Docs: man:openvpn(8)
          https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
          https://community.openvpn.net/openvpn/wiki/HOWTO
   Process: 9190 ExecStart=/usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn -
  Main PID: 9190 (code=exited, status=1/FAILURE)
    Status: "Pre-connection initialization successful"
      CPU: 16ms

Apr 04 07:43:33 tp-openvpn systemd[1]: openvpn@server.service: Main process exited, code=exited, status=1/FAILURE
Apr 04 07:43:33 tp-openvpn systemd[1]: openvpn@server.service: Failed with result 'exit-code'.
lines 1-13/13 (END)
```

Configuration

Nous copions le contenu de `easy-rsa` dans notre dossier `server` :

```
$ cp -r /usr/share/easy-rsa /etc/openvpn/server
```

```
root@tp-openvpn: /etc/openvpn
root@tp-openvpn /etc/openvpn# apt install easy-rsa
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
easy-rsa is already the newest version (3.1.0-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@tp-openvpn /etc/openvpn# cp -r /usr/share/easy-rsa /etc/openvpn/server
root@tp-openvpn /etc/openvpn# ls server
easy-rsa
root@tp-openvpn /etc/openvpn# ls server/easy-rsa/
easyrsa  openssl-easyrsa.cnf  vars.example  x509-types
root@tp-openvpn /etc/openvpn#
```

Nous créons notre infrastructure de clés publiques à l'aide de la commande suivante:

```
$ ./easy-rsa/easyrsa init-pki
$ ./easy-rsa/easyrsa build-ca
```

```
root@tp-openvpn: /etc/openvpn/server
root@tp-openvpn .../openvpn/server# ./easy-rsa/easyrsa init-pki
* Notice:

init-pki complete; you may now create a CA or requests.

Your newly created PKI dir is:
* /etc/openvpn/server/pki

* Notice:
IMPORTANT: Easy-RSA 'vars' file has now been moved to your PKI above.

root@tp-openvpn .../openvpn/server#
```

Note: Lors de l'exécution de la deuxième commande, indiquer autorité pour Common Name.

Nous effectuons ensuite la création de la clé et la demande de certificat pour Efrei et pour le client:

```
$ ./easy-rsa/easyrsa gen-req Efrei nopass
$ ./easy-rsa/easyrsa sign-req server Efrei
```

Nous obtenons le résultat suivant:

```
root@tp-openvpn: /etc/openvpn/server
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
    commonName          = Efrei

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes

Using configuration from /etc/openvpn/server/pki/bd372e7c/temp.a988a916
Enter pass phrase for /etc/openvpn/server/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'Efrei'
Certificate is to be certified until Jul  8 08:17:46 2026 GMT (825 days)

Write out database with 1 new entries
Database updated

* Notice:
Certificate created at: /etc/openvpn/server/pki/issued/Efrei.crt

root@tp-openvpn .../openvpn/server#
```

```
$ ./easy-rsa/easyrsa gen-req Client_VPN nopass
$ ./easy-rsa/easyrsa sign-req client ClientVPN
```

Nous obtenons le résultat suivant:

```
root@tp-openvpn: /etc/openvpn/server
root@tp-openvpn .../openvpn/server# ./easy-rsa/easyrsa sign-req client ClientVPN
* Notice:
Using Easy-RSA configuration from: /etc/openvpn/server/pki/vars

* Notice:
Using SSL: openssl OpenSSL 3.0.11 19 Sep 2023 (Library: OpenSSL 3.0.11 19 Sep 2023)

Easy-RSA error:

No request found for the input: 'ClientVPN'
Expected to find the request at: /etc/openvpn/server/pki/reqs/ClientVPN.req

Host: nix | Linux | /bin/bash

root@tp-openvpn .../openvpn/server#
```

Puis, nous créons la clé Diffie Hellman et la clé d'authentification TLS TA.key :

```
$ ./easy-rsa/easyrsa gen-dh
$ openvpn --genkey secret ta.key
```

Nous récupérons les différents fichiers :

```
$ scp root@192.168.1.252:/etc/openvpn/server/pki/ca.crt ./Desktop
$ scp root@192.168.1.252:/etc/openvpn/server/pki/issued/Client_VPN.crt ./Desktop
$ scp root@192.168.1.252:/etc/openvpn/server/pki/private/Client_VPN.key ./Desktop
$ scp root@192.168.1.252:/etc/openvpn/server/ta.key ./Desktop
```

Nous modifions maintenant le fichier de configuration `server.conf`, qui a maintenant le contenu suivant:

```
# PUBLIC_ADDRESS: vpn.Efrei.fr (used by openvpn-addclient)

port 1194
proto udp
dev tun

#keepalive 10 120

#persist-key
#persist-tun
#user nobody
#group nogroup

#chroot /etc/openvpn/easy-rsa/keys/crl.jail
#crl-verify /etc/openvpn/crl.pem

ca /etc/openvpn/pki/ca.crt
dh /etc/openvpn/pki/dh.pem
```

```

cert /etc/openvpn/pki/issued/Efrei.crt

#tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0
#key /etc/openvpn/easy-rsa/keys/private/server.key
#cert /etc/openvpn/easy-rsa/keys/issued/server.crt

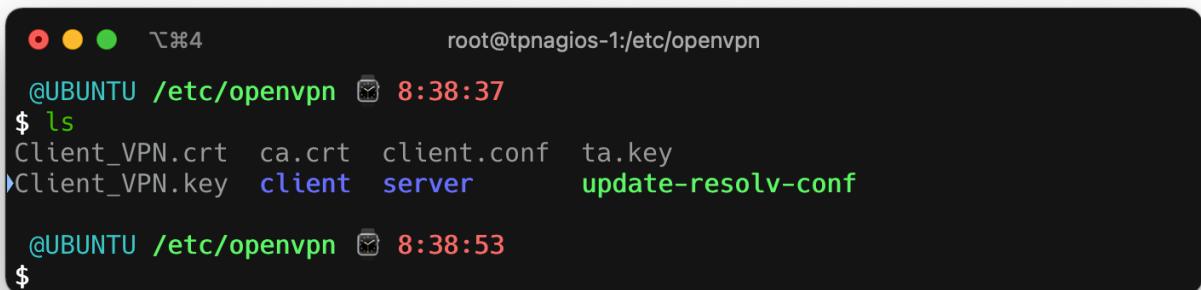
#ifconfig-poolPersist /var/lib/openvpn/server.ipp
#client-config-dir /etc/openvpn/server.ccd
#status /var/log/openvpn/server.log
#verb 4

# virtual subnet unique for openvpn to draw client addresses from
# the server will be configured with x.x.x.1
# important: must not be used on your network
#server 10.232.14.0 255.255.255.0
# push routes to clients to allow them to reach private subnets
#push "route 10.0.1.0 255.255.255.0"

plugin /usr/local/lib/openvpn-auth-ldap.so "/etc/openvpn/auth-ldap.conf"

```

Sur la machine cliente, nous copions tous les fichiers nécessaires à la connexion dans `/etc/openvpn`, et obtenons le résultat suivant:



```

root@tpnagios-1:/etc/openvpn
@UBUNTU /etc/openvpn 8:38:37
$ ls
Client_VPN.crt  ca.crt  client.conf  ta.key
Client_VPN.key  client  server      update-resolv-conf
@UBUNTU /etc/openvpn 8:38:53
$ 

```

Nous modifions notre fichier de configuration `client.conf` pour correspondre à notre configuration serveur:

```
vim client.conf
64
65 # Try to preserve some state across restarts.
66 persist-key
67 persist-tun
68
69 # If you are connecting through an
70 # HTTP proxy to reach the actual OpenVPN
71 # server, put the proxy server/IP and
72 # port number here. See the man page
73 # if your proxy server requires
74 # authentication.
75 ;http-proxy-retry # retry on connection failures
76 ;http-proxy [proxy server] [proxy port #]
77
78 # Wireless networks often produce a lot
79 # of duplicate packets. Set this flag
80 # to silence duplicate packet warnings.
81 ;mute-replay-warnings
82
83 # SSL/TLS parms.
84 # See the server config file for more
85 # description. It's best to use
86 # a separate .crt/.key file pair
87 # for each client. A single ca
88 # file can be used for all clients.
89 ca ca.crt
90 cert Client_VPN.crt
91 key Client_VPN.key
92
93 # Verify server certificate by checking that the
94 # certificate has the correct key usage set.
95 # This is an important precaution to protect against
96 # a potential attack discussed here:
97 # http://openvpn.net/howto.html#mitm
client.conf [+] 89,1 66%
-- INSERT --
```

Nous essayons ensuite de lancer le serveur à l'aide de la commande suivante depuis le dossier `/etc/openvpn` sur le serveur:

```
$ openvpn server.conf
```

Notre configuration est donc la suivante:

```
root@tp-openvpn: /etc/openvpn
root@tp-openvpn /etc/openvpn# grep . server.conf | grep -v '#'
port 1194
proto udp
dev tun
ca /etc/openvpn/pki/ca.crt
dh /etc/openvpn/pki/dh.pem
cert /etc/openvpn/pki/issued/Efrei.crt
plugin /usr/local/lib/openvpn-auth-ldap.so "/etc/openvpn/auth-ldap.conf"
root@tp-openvpn /etc/openvpn#
```

Malgré nos efforts, nous obtenons toujours une erreur dû à notre configuration TLS, pourtant à priori assez proche de celle du TP.

```
root@tp-openvpn /etc/openvpn# openvpn --tls-client server.conf
Options error: Unrecognized option or missing or extra parameter(s) in [CMD-LINE]:1: tls-client (2.6.3)
Use --help for more information.
root@tp-openvpn /etc/openvpn# openvpn server.conf
2024-04-04 08:47:01 WARNING: Ignoring option 'dh' in tls-client mode, please only include this in your server configuration
2024-04-04 08:47:01 DEPRECATION: No tls-client or tls-server option in configuration detected. OpenVPN 2.7 will remove the functionality to run a VPN without TLS. See the examples section in the manual page for examples of a similar quick setup with peer-fingerprint.
Options error: Parameter ca_file can only be specified in TLS-mode, i.e. where --tls-server or --tls-client is also specified.
Use --help for more information.
root@tp-openvpn /etc/openvpn# openvpn --tls-client server.^Cnf
root@tp-openvpn /etc/openvpn# openvpn --tls-server=1 server.conf
Options error: Unrecognized option or missing or extra parameter(s) in [CMD-LINE]:1: tls-server=1 (2.6.3)
Use --help for more information.
root@tp-openvpn /etc/openvpn# openvpn --tls-server 1 server.conf
Options error: Unrecognized option or missing or extra parameter(s) in [CMD-LINE]:1: tls-server (2.6.3)
Use --help for more information.
root@tp-openvpn /etc/openvpn# openvpn --tls-server=true server.conf
Options error: Unrecognized option or missing or extra parameter(s) in [CMD-LINE]:1: tls-server=true (2.6.3)
Use --help for more information.
root@tp-openvpn /etc/openvpn# vim server.c
root@tp-openvpn /etc/openvpn# vim server.conf
root@tp-openvpn /etc/openvpn# openvpn server.conf
2024-04-04 08:48:26 DEPRECATION: No tls-client or tls-server option in configuration detected. OpenVPN 2.7 will remove the functionality to run a VPN without TLS. See the examples section in the manual page for examples of a similar quick setup with peer-fingerprint.
Options error: Parameter cert_file can only be specified in TLS-mode, i.e. where --tls-server or --tls-client is also specified.
Use --help for more information.
root@tp-openvpn /etc/openvpn# vim server.conf
root@tp-openvpn /etc/openvpn# openvpn server.conf
2024-04-04 08:48:41 DEPRECATION: No tls-client or tls-server option in configuration detected. OpenVPN 2.7 will remove the functionality to run a VPN without TLS. See the examples section in the manual page for examples of a similar quick setup with peer-fingerprint.
2024-04-04 08:48:41 OpenVPN 2.6.3 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
```

La deuxième commande nous indique que l'utilisation d'un `ca.cert` nécessite l'utilisation du paramètre `--tls-server` d'après la commande `openvpn --help`. Le problème étant que, visiblement, ce paramètre n'est pas reconnu :

```
Options error: Unrecognized option or missing or extra parameter(s) in [CMD-LINE]:1: tls-client

# Alors que dans la doc, nous avons bien:
--tls-server      : Enable TLS and assume server role during TLS handshake.
# Qui ne semble pas nécessiter d'argument.
```

Étant à court d'idées et de temps, nous choisissons d'interrompre nos recherches ici.

Conclusion

En résumé, dans ce sprint, nous avons:

- Configuré les accès SSH à partir du LDAP
- Configuré les accès HTTP et HTTPS à un site web à partir d'un LDAP
- Configuré un serveur OpenVPN sur Proxmox de 4 façons différentes, dont une seule avec succès
- Connecté un client OpenVPN à partir de la configuration par défaut du conteneur LXC
- Configuré un serveur OpenVPN pour fonctionner avec des certificats
- Echoué au lancement du serveur OpenVPN et à la connexion d'un client à l'aide d'un certificat