

# Projet Sécurité de Systèmes d'information GL4

**Réalisé par :**

Hajji Roua

Saada Firas

Bouahaha Aymen

Hassoun Mohamed Karam

## La mise en place(setup)

Pour ce projet, nous commencerons par créer deux machines virtuelles avec Ubuntu.

machine 1 : machine serveur

ip : 192.168.100.4

hostname: ldap-server

machine 2 : machine client

ip : 192.168.100.5

hostname : ldap-client

## ▼ Partie 1 : Authentification avec OpenLDAP, SSH, Apache, OpenVPN:

### ▼ Section 1 - Configuration d'OpenLDAP

## ⚙️ Installations et configurations nécessaires

Package configuration

Configuring slapd

The DNS domain name is used to construct the base DN of the LDAP directory. For example, 'foo.example.org' will create the directory with 'dc=foo, dc=example, dc=org' as base DN.

DNS domain name:

insat.tn

<Ok>

```
root@ldap-server:/home/ldapsection# slapcat
dn: dc=insat,dc=tn
objectClass: top
objectClass: dcObject
objectClass: organization
o: gl4.tn
dc: insat
structuralObjectClass: organization
entryUUID: a9f54ebc-48b0-103e-92f5-83e048c2a66f
creatorsName: cn=admin,dc=insat,dc=tn
createTimestamp: 20240116114643Z
entryCSN: 20240116114643.393016Z#000000#000#000000
modifiersName: cn=admin,dc=insat,dc=tn
modifyTimestamp: 20240116114643Z

dn: cn=admin,dc=insat,dc=tn
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9R2Y3V0J5bDYzUUNoUVZ5M2VaOTlnb1MzU1ZBRGlCMlA=
structuralObjectClass: organizationalRole
entryUUID: a9f73380-48b0-103e-92f6-83e048c2a66f
creatorsName: cn=admin,dc=insat,dc=tn
createTimestamp: 20240116114643Z
entryCSN: 20240116114643.405466Z#000000#000#000000
modifiersName: cn=admin,dc=insat,dc=tn
modifyTimestamp: 20240116114643Z
```

```

root@ldap-server:/home/ldapsection# systemctl start slapd
root@ldap-server:/home/ldapsection# systemctl enable slapd
slapd.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable slapd
root@ldap-server:/home/ldapsection# systemctl enable slapd
slapd.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable slapd
root@ldap-server:/home/ldapsection# ldap whoami -x

Command 'ldap' not found, did you mean:

  command 'rdap' from snap rdap (2022.09.08+git)

See 'snap info <snapname>' for additional versions.

root@ldap-server:/home/ldapsection# ldapwhoami -x
anonymous

```

```

root@ldap-server: /home/ldapsection x root@ldap-server: ~ x
root@ldap-server:/home/ldapsection# ldapadd -x -D cn=admin,dc=insat,dc=tn -W -f
add_users\&groups.ldif
Enter LDAP Password:
adding new entry "ou=People,dc=example,dc=com"
ldap_add: Server is unwilling to perform (53)
        additional info: no global superior knowledge

root@ldap-server:/home/ldapsection#

```

### 1.1 - Configurer un serveur OpenLDAP avec au moins 2 utilisateurs et 2 groupes

Maintenant, tout est configuré pour ajouter nos groupes et utilisateurs

Créons le fichier LDIF suivant et appelez-le `add_users&groups.ldif`.

```

# Organizational Unit for People
dn: ou=People,dc=insat,dc=tn
objectClass: organizationalUnit
ou: People

# Organizational Unit for Groups
dn: ou=Groups,dc=insat,dc=tn
objectClass: organizationalUnit
ou: Groups

# Group 1: Miners
dn: cn=miners,ou=Groups,dc=insat,dc=tn
objectClass: posixGroup
cn: miners
gidNumber: 5000

# Group 2: Developers
dn: cn=developers,ou=Groups,dc=insat,dc=tn
objectClass: posixGroup
cn: developers
gidNumber: 5001

```

```

# Group 1: Miners
dn: cn=miners,ou=Groups,dc=insat,dc=tn
objectClass: posixGroup
cn: miners
gidNumber: 5000

# Group 2: Developers
dn: cn=developers,ou=Groups,dc=insat,dc=tn
objectClass: posixGroup
cn: developers
gidNumber: 5001

# User 1: Roua Hajji
dn: uid=roua,ou=People,dc=insat,dc=tn
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: roua
sn: Hajji
givenName: Roua
cn: Roua Hajji
displayName: Roua Hajji
uidNumber: 10000
gidNumber: 5000
userPassword: {CRYPT}x
gecos: Roua Hajji
loginShell: /bin/bash
homeDirectory: /home/roua

```

```
# User 2: Aymen Bouhaha
dn: uid=aymen,ou=People,dc=insat,dc=tn
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: aymen
sn: Bouhaha
givenName: Aymen
cn: Aymen Bouhaha
displayName: Aymen Bouhaha
uidNumber: 10001
gidNumber: 5001
userPassword: {CRYPT}x
gecos: Aymen Bouhaha
loginShell: /bin/bash
homeDirectory: /home/aymen
```

Exécutons ensuite cette commande pour ajouter le contenu :

```
ldapadd -x -D cn=admin,dc=insat,dc=tn -W -f add_users&gr
```

```
root@ldap-server:/home/ldapsection# ldapadd -x -D cn=admin,dc=insat,dc=tn -W -f
add_users\&groups.ldif
Enter LDAP Password:
adding new entry "ou=People,dc=insat,dc=tn"

adding new entry "ou=Groups,dc=insat,dc=tn"

adding new entry "cn=miners,ou=Groups,dc=insat,dc=tn"

adding new entry "cn=developers,ou=Groups,dc=insat,dc=tn"

adding new entry "uid=roua,ou=People,dc=insat,dc=tn"

adding new entry "uid=aymen,ou=People,dc=insat,dc=tn"
root@ldap-server:/home/ldapsection#
```

## 1.2 Ajout des informations et du certificat x509:

Tout d'abord, nous devons générer des certificats X509 pour chaque utilisateur. nous pouvons le faire en utilisant OpenSSL.

```
openssl genrsa > priv.key 2048
```

```

root@ldap-server:/home/ldapsection# openssl genrsa > priv.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
root@ldap-server:/home/ldapsection# openssl req -new -key priv.key > req.csr
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]:tunisia
string is too long, it needs to be no more than 2 bytes long
Country Name (2 letter code) [AU]:TN
State or Province Name (full name) [Some-State]:TUNIS
Locality Name (eg, city) []:ariana
Organization Name (eg, company) [Internet Widgits Pty Ltd]:gl4
Organizational Unit Name (eg, section) []:tech
Common Name (e.g. server FQDN or YOUR name) []:insat.tn
Email Address []:a@2.dk

```

```
openssl req -new -key priv2.key > req2.csr
```

```

root@ldap-server:/home/ldapsection# openssl req -new -key priv2.key > req2.csr
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]:TN
State or Province Name (full name) [Some-State]:tunis
Locality Name (eg, city) []:manar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:gl5
Organizational Unit Name (eg, section) []:ee
Common Name (e.g. server FQDN or YOUR name) []:efft
Email Address []:fff

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:0001
An optional company name []:

```

```
openssl x509 -req -days 365 < req.csr -signkey priv.key :
```

```

root@ldap-server:/home/ldapsection# openssl x509 -req -days 365 < req.csr -sign
key priv.key > cert.crt
Signature ok
subject=C = TN, ST = TUNIS, L = ariana, O = gl4, OU = tech, CN = insat.tn, email
Address = a@2.dk
Getting Private key

```



Maintenant, pour ajouter les certificats, nous devons convertir le certificat PEM au format DER ,ensuite, encodons en base64 le certificat DER, (mais sans sauts de ligne (l'option -A )) :

```
openssl x509 < cert.crt -outform der > cert.der
openssl base64 -A < cert.der > cert.der.base64
```

```
root@ldap-server:/home/ldapsection# ls
'add_users&groups.ldif'  cert.crt  priv.key  req.csr
cert2.crt               priv2.key req2.csr
root@ldap-server:/home/ldapsection# openssl x509 < cert.crt -outform der > cert
.der
root@ldap-server:/home/ldapsection# openssl x509 < cert2.crt -outform der > cer
t2.der
root@ldap-server:/home/ldapsection# openssl base64 -A < cert.der > cert.der.b64
root@ldap-server:/home/ldapsection# openssl base64 -A < cert2.der > cert2.der.b
64
root@ldap-server:/home/ldapsection#
```

Nous pouvons maintenant ajouter le certificat DER codé en base64 au répertoire LDAP. Cependant, nous ne pouvons pas utiliser la directive `file://` dans le fichier LDIF. Nous devons insérer le certificat encodé en base64 directement dans le fichier LDIF.

```
root@ldap-server:/home/ldapsection# cat aymen.ldif
dn: uid=aymen,ou=People,dc=insat,dc=tn
changetype: modify
add: userCertificate;binary
userCertificate;binary::MIIDXTCCAkUCFBJKDP3roeArEzihAr8y5l06H4QwMA0GCSqGSIb3DQE
BCwUAMGsx CzAJBgNVBAYTALROMQ4wDAYDVQQIDAV0dW5pczEOMAwGA1UEBwwFbWYuYXIXDDAKBgNVBA
oMA2dsNTElMAKGA1UECwwCZWUxDALBgNVBAMMBGVmZnQxEjAQBgkqhkiG9w0BCQEWA2ZmZjAeFw0yN
DAXMTYxMjA3MDZaFw0yNTAxMTUxMjA3MDZaMGsx CzAJBgNVBAYTALROMQ4wDAYDVQQIDAV0dW5pczEO
MAwGA1UEBwwFbWYuYXIXDDAKBgNVBAoMA2dsNTElMAKGA1UECwwCZWUxDALBgNVBAMMBGVmZnQxEjA
QBgkqhkiG9w0BCQEWA2ZmZjCCASIwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMP8nPILW6cpBJ
VpWI/iplxCJ1w9LKhyLPL6C/GyQy/AnagwYbGib1G31azvdlFA+FtRxlw7KzWZM4adXIYM6gmKlHz7J
lgXfb4Dq+GSfKcJ2EbPkkhJ3H16rLIFF0ogXHCzssj5wvceobM8NDKtUJnxZ2cGQ7T1In22xoubTrm0
eHd0DBxY9UaqcySYjZbcch9M3X0ln+UTSI40TpYS6K7s0cbUYuvk4ZaJ8P1EPMyslgdXfQf4EiWgd0
FTZldJ4XfMpOXsUkpPHF6qJ0y/XCJ+ZP/x8lvYmDalh3iVixdpAjmqkJPLTfUmTXUwZS5zgK4TW+UPr
DdZs3XSejRzrMCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAvfCN40KgPV9ug78Sqmgv6IwwrU5tHfjy2
1mjjsNR0wTp2th9j70BEctQ5EAIeAejIfvYTglhIzsVsXTcDbrrKA3QBvzfE7teZuLj2iQxyekCzm0
jMb+nw0tF+Kk8NDPgQ6WId0j8tjoIGjPrx123SbwuM/j0fK4q0H4nw6IK29ZUESn/8ecvW8xImBpw4S
yJovuzgIuJN1lvReAS9tvb2+tjcWWQ7LeSDiZKhTxzeb/BBGz9dEWIgUAZsS3mvrQaBJeuPuiBu4ozP
at7tjwplhYydoH+F6wC8tLAJLjwFANpqnywdzIJK3P/8onK59K2h9IK1Q9oROY1SX4pG3XoA==
```

Ensuite, utilisons la commande `ldapmodify` pour traiter le fichier LDIF :

```
root@ldap-server:/home/ldapsection# nano aymen.ldif
root@ldap-server:/home/ldapsection# ldapmodify -x -D cn=admin,dc=insat,dc=tn -W
-f aymen.ldif
Enter LDAP Password:
modifying entry "uid=aymen,ou=People,dc=insat,dc=tn"
```

### 1.3 - Authentification avec succès sur le serveur OpenLDAP.

Nous devons d'abord définir un nouveau mot de passe pour chaque utilisateur (roua + aymen)

```
root@ldap-server:/home/ldapsection# ldappasswd -H ldapi:/// -x -D cn=admin,dc=insat,dc=tn -W -S uid=aymen,ou=People,dc=insat,dc=tn
New password:
Re-enter new password:
Enter LDAP Password:
root@ldap-server:/home/ldapsection#
```

Ensuite, nous nous assurons que le pare-feu est configuré

```
root@ldap-server:/home/ldapsection# sudo ufw allow 389
Skipping adding existing rule
Skipping adding existing rule (v6)
root@ldap-server:/home/ldapsection# ufw reload
Firewall reloaded
root@ldap-server:/home/ldapsection# sudo ufw status
Status: active
```

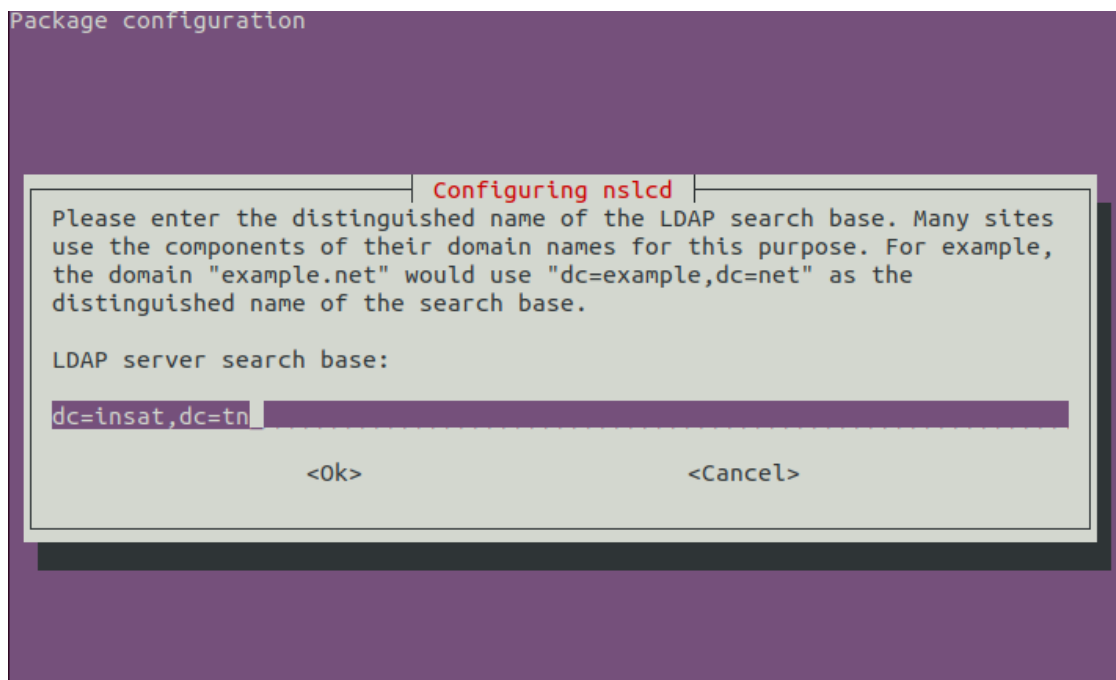
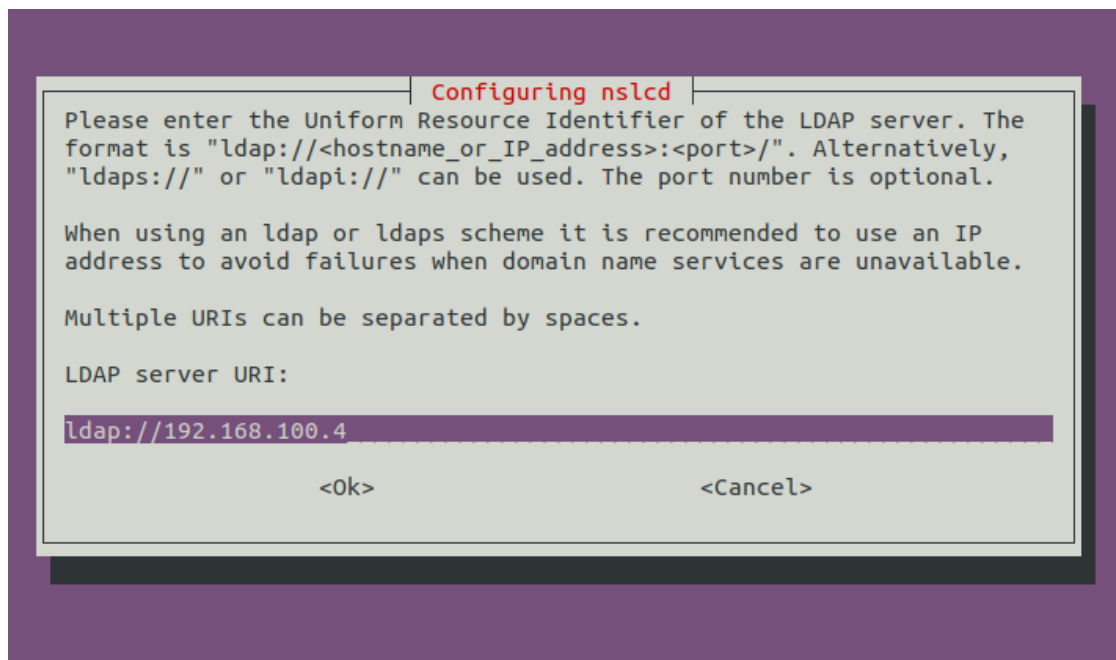
To	Action	From
--	-----	----
389	ALLOW	Anywhere
389 (v6)	ALLOW	Anywhere (v6)

Maintenant, nous allons au CLIENT et installons les packages suivants en utilisant la commande suivante

```
apt install libnss-ldapd libpam-ldapd ldap-utils
```

puis on rentre l'IP de la machine serveur





**puis on redémarre les services en utilisant la commande suivante:**

```
systemctl restart nscd nslcd
```

**Nous pouvons maintenant nous authentifier auprès d'un utilisateur depuis le serveur LDAP comme suit :**

```
vboxuser@ldapclient:~$ su -l aymen
Password:
su: warning: cannot change directory to /home/aymen: No such file or directory
aymen@ldapclient:/home/vboxuser$ exit
logout
vboxuser@ldapclient:~$ su -l roua
Password:
su: warning: cannot change directory to /home/roua: No such file or directory
roua@ldapclient:/home/vboxuser$
```

## 1.4 - Tester la partie sécurisée de LDAP avec ldaps et décrire les différents avantages.

Pour accomplir cette tâche, nous utiliserons Transport Layer Security (TLS). Sera notre propre autorité de certification (CA), puis créera et signera notre certificat de serveur LDAP en tant que cette autorité de certification.

Nous installons d'abord les packages suivants

```
sudo apt install gnutls-bin ssl-cert
```

Ensuite, nous créons une clé privée

```
root@ldap-server:/home/ldapsection# sudo certtool --generate-privkey --bits 4096 --outfile /etc/ssl/private/mycakey.com
** Note: You may use '--sec-param High' instead of '--bits 4096'
Generating a 4096 bit RSA private key...
```

Créons le `/etc/ssl/ca.info` pour définir l'autorité de certification

```
root@ldap-server:/etc/ssl# cat ca.info
cn = Insat
ca
cert_signing_key
expiration_days = 3650
```

Ensuite, nous créons le certificat CA auto-signé

```
root@ldap-server:/etc/ssl/private# certtool --generate-self-signed --load-privkey /etc/ssl/private/mycakey.pem --template /etc/ssl/ca.info --outfile /usr/local/share/ca-certificates/mycacert.crt
Generating a self signed certificate...
X.509 Certificate Information:
```

Exécutons `update-ca-certificates` pour ajouter le nouveau certificat d'autorité de certification à la liste des autorités de certification de confiance.

```

root@ldap-server:/etc/ssl/private# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt, it does not contain exactly one c
ertificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.

```

Maintenant, nous créons une clé privée pour le serveur

```

root@ldap-server:/etc/ldap# certtool --generate-privkey --bits 2048 --outfile /
etc/ldap/ubuntu-server_slapd_key.pem
** Note: You may use '--sec-param Medium' instead of '--bits 2048'
Generating a 2048 bit RSA private key...

```

Créons le fichier d'informations `/etc/ssl/kali-server.info` contenant :

```

root@ldap-server:/etc/ssl# cat ubuntu-server.info
organisation= Insat
cn= kali-server.insat.com
tls_www_server
encryption_key
signing_key
expiration_days= 365

```

Créons le certificat du serveur :

```

root@ldap-server:/etc/ssl# certtool --generate-certificate --load-privkey /etc
/ldap/ubuntu-server_slapd_key.pem --load-ca-certificate /etc/ssl/certs/mycacer
t.pem --load-ca-privkey /etc/ssl/private/mycakey.pem --template /etc/ssl/ubun
tu-server.info --outfile /etc/ldap/ubuntu-server_slapd_cert.pem
Warning: skipping unknown option 'organisation'
Generating a signed certificate...
X.509 Certificate Information:
    Version: 3
    Serial Number (hex): 4b0cf74c1a201db501f4c0ef0e7689a20fd1ac76
    Validity:
        Not Before: Thu Jan 18 23:14:00 UTC 2024

```

Ajustons les autorisations et la propriété :

```

root@ldap-server:/etc/ssl# chgrp openssl /etc/ldap/ubuntu-server_slapd_key.pem
root@ldap-server:/etc/ssl# chmod 0640 /etc/ldap/ubuntu-server_slapd_key.pem

```

**Notre serveur est désormais prêt à accepter la nouvelle configuration TLS.**

Créons le fichier `certinfo.ldif` avec le contenu suivant

```

root@ldap-server:/etc/ssl# cat certinfo.ldif
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/mycacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/ubuntu-server_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/ubuntu-server_slapd_key.pem

```

Utilisons la commande `ldapmodify` pour informer `slapd` de notre travail TLS via la base de données slapd-config :

```

root@ldap-server:/etc/ssl# ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
ldapmodify: wrong attributeType at line 3, entry "cn=config "

```

Modifions `/etc/default/slapd` et incluez `ldaps:///` dans `SLAPD_SERVICES` comme ci-dessous :

```
SLAPD_SERVICES="ldap:/// ldapi:/// ldaps://"
```

```
SLAPD_SERVICES="ldap:/// ldapi:/// ldaps://"
```

Et redémarrons `slapd` avec : `sudo systemctl restart slapd`

Nous sommes maintenant prêts à tester notre LDAPS

```

root@ldap-server:/etc/ssl# ldapwhoami -x -H //kali-server.insat.com

```

## ▼ Section 2 - Authentification SSH

### 2.1 Activez l'authentification SSH via OpenLDAP.

Nous modifions d'abord le schéma utilisateur LDAP en créant `openssh-lpk.ldif` puis nous appliquons la modification.

```

root@ldap-server:/home# cat openssl-lpk.ldif
dn: cn=openssl-lpk,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: openssl-lpk
olcAttributeTypes: ( 1.3.6.1.4.1.24552.500.1.1.1.13 NAME 'sshPublicKey'
    DESC 'MANDATORY: OpenSSH Public key'
    EQUALITY octetStringMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
olcObjectClasses: ( 1.3.6.1.4.1.2.24552.500.1.1.2.0 NAME 'ldapPublicKey' SUP to
p
    DESC 'MANDATORY: OpenSSH LPK objectclass'
    MAY ( sshPublicKey $ uid )
    )
root@ldap-server:/home# ldapadd -Y EXTERNAL -H ldapi:/// -f openssl-lpk.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
a Help new entry "cn=openssl-lpk,cn=schema,cn=config"
ldap_add: Other (e.g., implementation specific) error (80)
    additional info: olcAttributeTypes: Duplicate attributeType: "1.3.6.1.4
.1.24552.500.1.1.1.13"

```

```

root@ldap-server:/home# ldapsearch -H ldapi://localhost -x -s base -b "cn=subsch
ema" objectclasses | grep ldapPublicKey
objectClasses: ( 1.3.6.1.4.1.2.24552.500.1.1.2.0 NAME 'ldapPublicKey' DESC 'MA
root@ldap-server:/home#

```

Maintenant, nous créons notre paire de clés sur la machine client :

```

aymen@ldapclient:/home/vboxuser$ sudo ls /root
[sudo] password for aymen:
snap vboxpostinstall.sh
aymen@ldapclient:/home/vboxuser$ sudo ssh-keygen -t rsa -b 4096 -C "aymen"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/.ssh/id_rsa
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Uqs3aKpFWGtXVQpSKq94fcAckBc8+FjYnMuJOjdpqTY aymen
The key's randomart image is:
+----[RSA 4096]-----+
|      .Bo+o ...      |
|      +.Oo o .       |
|      .oBo+o .       |
|      Eo+*=+ .       |
|      ..o+oB S       |
|      o+=+ =         |
|      .++.= +        |
|      o o o .        |
|      ...            |
+----[SHA256]-----+

```

On installe openssl-server sur la machine serveur avec la comande suivante:

```
sudo apt install openssh-server
```

```
root@ldap-server:/home# apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.11).
0 upgraded, 0 newly installed, 0 to remove and 171 not upgraded.
root@ldap-server:/home#
```

Partageons ensuite la clé publique avec le serveur

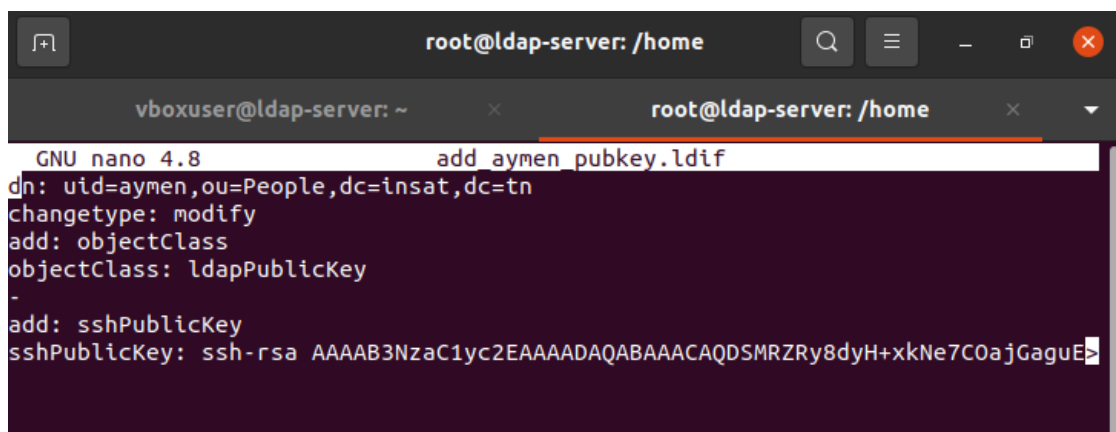
```
root@ldapclient:/etc# ssh-copy-id -i /root/.ssh/id_rsa.pub vboxuser@192.168.100.4
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
vboxuser@192.168.100.4's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'vboxuser@192.168.100.4'"
and check to make sure that only the key(s) you wanted were added.
```

```
root@ldap-server:/home/vboxuser# ls .ssh
authorized_keys
```

Modifions le schéma pour stocker la clé publique ssh avec le serveur, puis on ajoute la clé à l'entrée utilisateur



```
root@ldap-server: /home
GNU nano 4.8 add_aymen_pubkey.ldif
dn: uid=aymen,ou=People,dc=insat,dc=tn
changetype: modify
add: objectClass
objectClass: ldapPublicKey
-
add: sshPublicKey
sshPublicKey: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDSMRZry8dyH+xkNe7C0ajGaguE
```



```
root@ldap-server:/home# ldapmodify -x -D "cn=admin,dc=insat,dc=tn" -W -f add_aymen_pubkey.ldif
Enter LDAP Password:
modifying entry "uid=aymen,ou=People,dc=insat,dc=tn"
```

On configure maintenant une nouvelle machine en tant que client LDAP (comme nous l'avons fait avec le serveur). Cette machine est celle que nous voulons authentifier via LDAP. Ensuite, on installe `openssh-server`. Après cela, on modifie `/etc/ssh/sshd_config` et édite `/etc/pam.d/sshd`. Puis on essaye l'authentification

## 2.2 - Restrict access to certain groups Restreignez l'accès SSH aux utilisateurs du groupe approprié dans OpenLDAP.

### Étape 1 : Configurez PAM pour utiliser le contrôle d'accès.

1. Installons le module `pam_access` s'il n'est pas déjà installé. Ce module est généralement inclus dans l'installation PAM par défaut.
2. Modifions le fichier de configuration PAM SSH, `/etc/pam.d/sshd`, sur la machine (le serveur SSH) pour inclure `pam_access`. Ajoutons la ligne suivante à la section authentication :

```
auth required pam_access.so
```

### Étape 2 : Définir les règles d'accès.

1. Modifions le fichier de contrôle d'accès `/etc/security/access.conf` sur la machine. Ce fichier définit les règles pour autoriser ou refuser l'accès aux services.
2. Ajoutons une règle pour autoriser l'accès à un groupe LDAP spécifique. Par exemple, si vous souhaitez autoriser l'accès uniquement aux utilisateurs du groupe LDAP `sshusers`, ajoutez la ligne suivante :

```
+ : (sshusers) : ALL
```

Cette ligne signifie "autoriser les membres du groupe `sshusers` à accéder depuis TOUTES les emplacements".

### Étape 3 : Configurez NSS pour reconnaître les groupes LDAP.

Ensure that the NSS configuration on `Machine2` is set up to recognize LDAP groups by editing `/etc/nsswitch.conf` and verifying that the `group` line includes `ldap` :

Assurons que la configuration NSS sur la machine est configurée pour reconnaître les groupes LDAP en modifiant `/etc/nsswitch.conf` et en vérifiant que la ligne `group` inclut `ldap` :

```
group: files ldap
```

## ▼ Section 3 - Intégration d'Apache

### 3.1 - Configuring apache to user LDAP authentication

Installons Apache2

```
apt-get install apache2
```

```
apt-get install apache2-utils
```

On active le module `authnz_ldap` pour Apache, permettant au serveur d'utiliser LDAP pour l'authentification et l'autorisation.

```
sudo a2enmod authnz_ldap
```

On active le module `cache_disk` pour Apache, qui est utilisé pour mettre en cache du contenu sur le disque afin d'améliorer les performances.

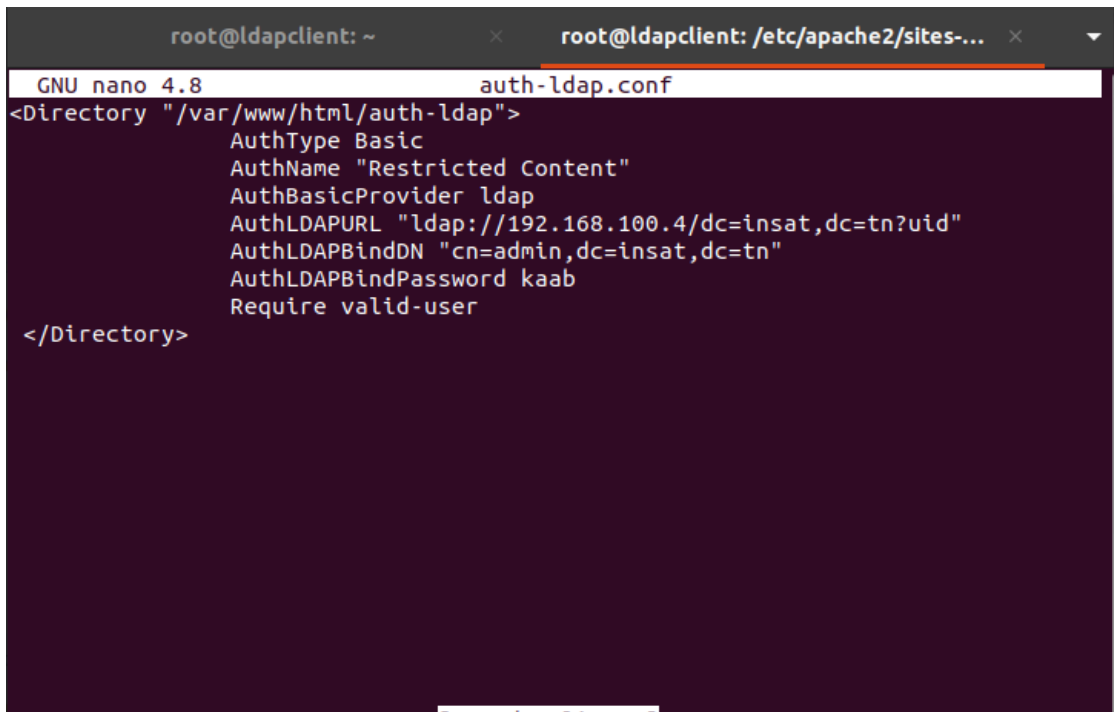
```
sudo a2enmod cache_disk
```

On active le module `rewrite` pour Apache, offrant la possibilité de manipuler les URL et d'effectuer des redirections ou des réécritures d'URL en fonction de règles définies.

```
sudo a2enmod authnz_ldap
sudo a2enmod cache_disk
sudo a2enmod rewrite
```

```
root@ldapclient:/etc/apache2/sites-available# a2enmod ldap authnz_ldap
Module ldap already enabled
Considering dependency ldap for authnz_ldap:
Module ldap already enabled
Module authnz_ldap already enabled
```

On modifie le fichier de configuration `auth-ldap.conf`



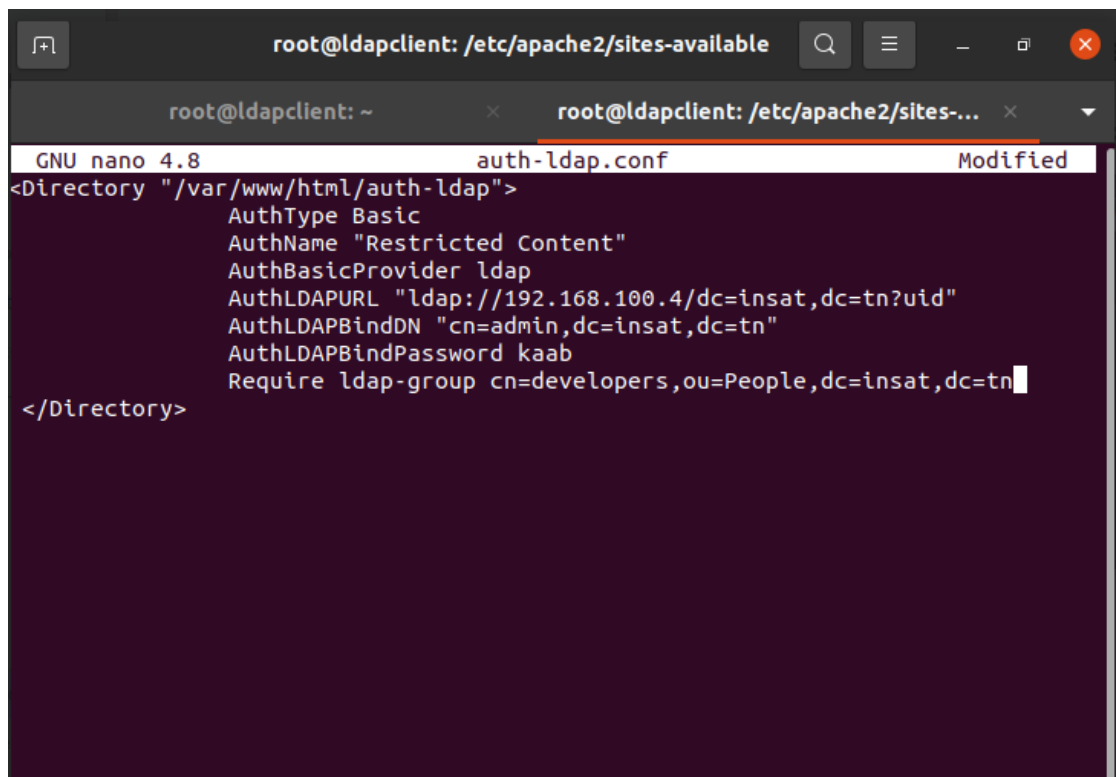
The screenshot shows a terminal window with two tabs. The active tab is titled 'root@ldapclient: /etc/apache2/sites-...' and contains the GNU nano 4.8 editor editing the file 'auth-ldap.conf'. The editor shows the following XML configuration for a directory:

```
<Directory "/var/www/html/auth-ldap">
    AuthType Basic
    AuthName "Restricted Content"
    AuthBasicProvider ldap
    AuthLDAPURL "ldap://192.168.100.4/dc=insat,dc=tn?uid"
    AuthLDAPBindDN "cn=admin,dc=insat,dc=tn"
    AuthLDAPBindPassword kaab
    Require valid-user
</Directory>
```

Maintenant, nous devons nous authentifier avec un utilisateur provenant de LDAP.

### 3.2 - Restrict access to manager group

Cette modification permet d'autoriser l'authentification uniquement avec les utilisateurs du groupe de "developers".



The screenshot shows a terminal window with a dark background. At the top, the title bar indicates the user is root@ldapclient in the directory /etc/apache2/sites-available. Below the title bar, there are two tabs: 'root@ldapclient: ~' and 'root@ldapclient: /etc/apache2/sites-...'. The active tab shows the nano 4.8 editor editing the file 'auth-ldap.conf'. The file content is as follows:

```
<Directory "/var/www/html/auth-ldap">
  AuthType Basic
  AuthName "Restricted Content"
  AuthBasicProvider ldap
  AuthLDAPURL "ldap://192.168.100.4/dc=insat,dc=tn?uid"
  AuthLDAPBindDN "cn=admin,dc=insat,dc=tn"
  AuthLDAPBindPassword kaab
  Require ldap-group cn=developers,ou=People,dc=insat,dc=tn
</Directory>
```

### 3.3 - Test for a group with and without access permission

## ▼ Section 4 - Mise en place d'OpenVPN

### 4.1 - Setup OpenVPN and enable LDAP authentication

Commençons par l'installation de OpenVPN sur 2 machines.

```
apt install openvpn
```

Nous pouvons maintenant créer un tunnel non sécurisé.

```

root@ldap-server:/home# openvpn --dev tun --remote 192.168.100.5 --ifconfig 1.1.1.2 1.1.1.1
Fri Jan 19 05:55:09 2024 disabling NCP mode (--ncp-disable) because not in P2MP client or server mode
Fri Jan 19 05:55:09 2024 OpenVPN 2.4.12 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Aug 21 2023
Fri Jan 19 05:55:09 2024 library versions: OpenSSL 1.1.1f 31 Mar 2020, LZO 2.10
Fri Jan 19 05:55:09 2024 ***** WARNING *****: All encryption and authentication features disabled -- All data will be tunneled as clear text and will not be protected against man-in-the-middle changes. PLEASE DO RECONSIDER THIS CONFIGURATION!
Fri Jan 19 05:55:09 2024 TUN/TAP device tun0 opened
Fri Jan 19 05:55:09 2024 /sbin/ip link set dev tun0 up mtu 1500
Fri Jan 19 05:55:09 2024 /sbin/ip addr add dev tun0 local 1.1.1.2 peer 1.1.1.1
Fri Jan 19 05:55:09 2024 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.100.5:1194
Fri Jan 19 05:55:09 2024 UDP link local (bound): [AF_INET][undef]:1194
Fri Jan 19 05:55:09 2024 UDP link remote: [AF_INET]192.168.100.5:1194
Fri Jan 19 05:55:19 2024 Peer Connection Initiated with [AF_INET]192.168.100.5:1194
Fri Jan 19 05:55:19 2024 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
Fri Jan 19 05:55:19 2024 Initialization Sequence Completed

```

```

root@ldapclient:/etc/apache2/sites-available# openvpn --dev tun --ifconfig 1.1.1.1 1.1.1.2
Fri Jan 19 05:53:46 2024 disabling NCP mode (--ncp-disable) because not in P2MP client or server mode
Fri Jan 19 05:53:46 2024 OpenVPN 2.4.12 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Aug 21 2023
Fri Jan 19 05:53:46 2024 library versions: OpenSSL 1.1.1f 31 Mar 2020, LZO 2.10
Fri Jan 19 05:53:46 2024 ***** WARNING *****: All encryption and authentication features disabled -- All data will be tunneled as clear text and will not be protected against man-in-the-middle changes. PLEASE DO RECONSIDER THIS CONFIGURATION!
Fri Jan 19 05:53:46 2024 TUN/TAP device tun0 opened
Fri Jan 19 05:53:46 2024 /sbin/ip link set dev tun0 up mtu 1500
Fri Jan 19 05:53:46 2024 /sbin/ip addr add dev tun0 local 1.1.1.1 peer 1.1.1.2
Fri Jan 19 05:53:46 2024 Could not determine IPv4/IPv6 protocol. Using AF_INET
Fri Jan 19 05:53:46 2024 UDPv4 link local (bound): [AF_INET][undef]:1194
Fri Jan 19 05:53:46 2024 UDPv4 link remote: [AF_UNSPEC]
Fri Jan 19 05:55:09 2024 Peer Connection Initiated with [AF_INET]192.168.100.4:1194
Fri Jan 19 05:55:10 2024 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
Fri Jan 19 05:55:10 2024 Initialization Sequence Completed

```

Pour l'authentification LDAP, nous devons d'abord créer un utilisateur système en lecture seule sur notre serveur LDAP afin de l'utiliser ultérieurement pour l'authentification.

We create an `ldif` file to add `system ou` and `readonly user`

Nous créons un fichier LDIF pour ajouter `system ou` et `readonly user`

```

root@ldap-server:/home/vboxuser# ldapadd -x -D "cn=admin,dc=insat,dc=tn" -W -f
readonly.ldif
Enter LDAP Password:
adding new entry "ou=system,dc=insat,dc=tn"

adding new entry "cn=readonly,ou=system,dc=insat,dc=tn"

```

Ensuite, nous lui attribuons la permission en lecture seule en créant un fichier de configuration.

```

root@ldap-server:/home/vboxuser# ldapmodify -Y EXTERNAL -H ldapi:/// -f grant_r
eadonly_access.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0

```

Maintenant, nous pouvons commencer à travailler sur le serveur OpenVPN.

Nous commençons par installer le plugin `openvpn-auth-ldap` .

```
apt install openvpn-auth-ldap
```

Nous créons un nouveau répertoire `/etc/openvpn/auth` et y copions le fichier `auth-ldap.conf`.

```

root@ldap-server:/etc/openvpn# cp /usr/share/doc/openvpn-auth-ldap/examples/auth-
ldap.conf /etc/openvpn/auth/ldap.conf

```

```

<LDAP>
# LDAP server URL
URL ldap://ldap1.example.org

# Bind DN (If your LDAP server doesn't support anonymous binds)
# BindDN uid=Manager,ou=People,dc=example,dc=com

# Bind Password
# Password SecretPassword

# Network timeout (in seconds)
Timeout 15

# Enable Start TLS
TLSEnable yes

# Follow LDAP Referrals (anonymously)
FollowReferrals yes

# TLS CA Certificate File
TLSCACertFile /usr/local/etc/ssl/ca.pem

```



Nous ajoutons nos fichiers de configuration à `server.conf`

```
root@ldap-server:/etc/openvpn/auth# echo "plugin /usr/lib/openvpn-auth-ldap.so
/etc/openvpn/auth/ldap.conf" >> /etc/openvpn/server/server.conf
```

Maintenant, nous pouvons tester l'authentification.

## ▼ Partie 2: Gestion des Services Réseau avec DNS

### 1. Installation de BIND sur les serveurs DNS

nous avons utilisé une machine serveur et une machine client

Ensuite, sur chaque machine, on installe BIND

```
sudo apt install bind9 bind9utils bind9-doc
```

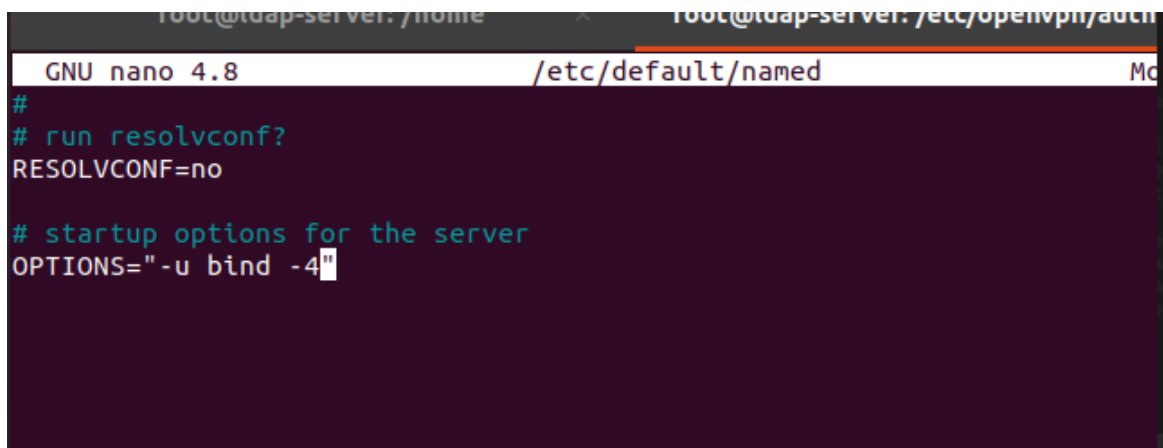
On définit BIND en mode IPv4.

Sur les deux serveurs, modifiez le fichier de paramètres par défaut nommé.

```
sudo nano /etc/default/named
```

Ajoutez `-4` à la fin du paramètre OPTIONS : `OPTIONS="-u bind -4"`

.



```
root@ldap-server: /home
root@ldap-server: /etc/openvpn/auth

GNU nano 4.8 /etc/default/named Mo
#
# run resolvconf?
RESOLVCONF=no

# startup options for the server
OPTIONS="-u bind -4"
```

On redémarre BIND pour implémenter les modifications, avec la commande suivante:

```
sudo systemctl restart bind9
```

## 2. Configuration du serveur DNS principal

Nous commencerons par configurer le fichier `named.conf.options`.

Côté serveur, on modifie le fichier `named.conf.options` :

```
sudo nano /etc/bind/named.conf.options
```

```
acl "trusted" {  
    192.168.100.4;  
    192.168.100.5  
};
```

On peut maintenant modifier le bloc d'options.

```
options {  
    directory "/var/cache/bind";  
    . . .  
}
```

Code complet

```
acl "trusted" {  
    192.168.100.4 ;  
    192.168.100.5  
};  
options {  
    directory "/var/cache/bind";  
  
    // If there is a firewall between you and nameservers you want  
    // to talk to, you may need to fix the firewall to allow multiple  
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113  
  
    // If your ISP provided one or more IP addresses for stable  
    // nameservers, you probably want to use them as forwarders.  
    // Uncomment the following block, and insert the addresses replacing  
    // the all-0's placeholder.  
  
    forwarders {  
        8.8.8.8;  
        8.8.4.4;  
    };  
    //=====  
    // If BIND logs error messages about the root key being expired.
```

Seuls nos propres serveurs (ceux de confiance) pourront interroger le serveur DNS pour les domaines extérieurs.

## Maintenant on passe à la configuration du fichier local

Sur coté serveur , on modifie fichier named.conf.local :

```
GNU nano 4.8 /etc/bind/named.conf.local Modified
//
// 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 "insat.tn" {
    type primary;
    file "etc/bind/zones/db.insat.tn" ;
    allow-transfer {192.168.100.5} ;
};
zone "192.168.in-addr.arpa" {
    type primary;
    file "etc/bind/zones/db.192.168" ;
    allow-transfer {192.168.100.5} ;
};
```

## Maintenant on passe à la création du fichier de zone forward:

Le fichier de zone de transfert est l'endroit où vous définissez les enregistrements DNS pour les recherches DNS directes.

Créons le répertoire dans lequel résideront les fichiers de zone.

Selon la configuration named.conf.local, ils doivent être dans `/etc/bind/zones` :

```
sudo mkdir /etc/bind/zones
```

On copie le fichier de zone db.localle à sa place avec les commandes suivantes :

```
sudo cp /etc/bind/db.local /etc/bind/zones/db.insat.tn
```

Modifions maintenant le fichier de zone de transfert :

```
sudo nano /etc/bind/zones/db.insat.tn
```

Code complet final: contenu de `db.insat.tn`

```
GNU nano 4.0 /etc/bind/zones/db.insat.tn
;
; BIND data file for local loopback interface
;
$TTL      604800
@          IN      SOA      server.insat.tn. admin.insat.tn. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
; name servers - NS records
; IN      NS      server.insat.tn.
; name servers - A records
server.insat.tn.      IN      A          192.168.100.4
; 192.168.100.0/16 - A records
client.insat.tn.      IN      A          192.168.100.5

Save modified buffer?
Y Yes
N No      ^C Cancel
```

## On passe maintenant à la création du fichier de zone reverse

Les fichiers de zone reverse sont l'endroit où vous définissez les enregistrements DNS PTR pour les recherches DNS inversées.

Côté serveur, pour chaque zone inversée spécifiée dans le fichier

`named.conf.local`, créez un fichier de zone inversée. Nous baserons nos exemples de fichiers de zone inversée sur l'exemple de fichier de zone `db.127`. BIND utilise ce fichier pour stocker des informations pour l'interface de bouclage locale ; 127 est le premier octet de l'adresse IP qui représente localhost (127.0.0.1).

On copie ce fichier à l'emplacement approprié en remplaçant le nom du fichier de destination pour qu'il corresponde à la définition de la zone inversée :

```
sudo cp /etc/bind/db.127 /etc/bind/zones/db.192.168
```

Editons le fichier de zone inversée qui correspond à la zone inversée définie dans `named.conf.local` :

```
sudo nano /etc/bind/zones/db.192.168
```

Code complet final: contenu de `db.insat.tn`

```

;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      insat.tn  admin.insat.tn (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
; name servers
;         IN      NS       server.insat.tn.
; PTR Records
100.4     IN      PTR      server.insat.tn      ; 192.168.100.4
100.5     IN      PTR      client.insat.tn      ; 192.168.100.4=5

```

## Passons maintenant a la vérification de la syntaxe de configuration BIND

On vérifie la syntaxe des fichiers `named.conf*` :

```

root@ldap-server:/etc/openvpn/auth# sudo named-checkconf
root@ldap-server:/etc/openvpn/auth#

```

pour la vérification de la configuration de la zone de transfert insat.tn , on suit cette commande

```

root@ldap-server:/etc/openvpn/auth# named-checkzone insat.tn /etc/bind/zones/db.insat.tn
zone insat.tn/IN: loaded serial 2
OK

```

pour la vérification de la configuration de la zone inversée 192.168.in-addr.arpa, on suit cette commande

```

root@ldap-server:/etc/openvpn/auth# named-checkzone 192.168 /etc/bind/zones/db.192.168
zone 192.168/IN: loaded serial 1
OK

```

## Redémarrage de BIND

```
sudo systemctl restart bind9
```

#### 4. Configuration des clients DNS

On recherche les périphériques associés au réseau privé.

```
ip address show to 192.168.0.0/16
```

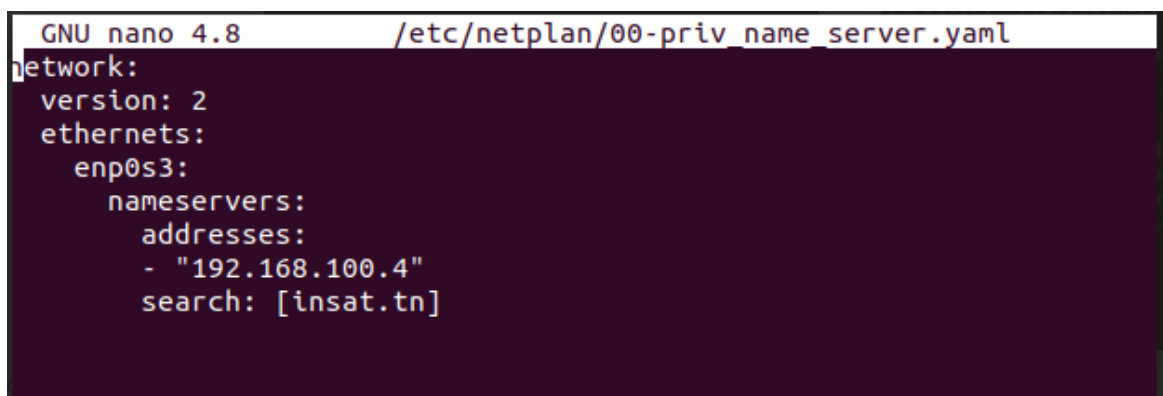
On crée un nouveau fichier `00-private-nameservers.yaml` dans `/etc/netplan` :

```
sudo nano /etc/netplan/00-private-nameservers.yaml
```

On modifie l'interface du réseau privé avec l'adresse de notre serveur côté, et la zone DNS :



```
GNU nano 4.8 /etc/netplan/00-priv_name_server.yaml
network:
  version:2
  ethernet:
    enp0s8:
      nameservers:
        addresses:
          - 192.168.100.4
        search: [ insat.tn]
```



```
GNU nano 4.8 /etc/netplan/00-priv_name_server.yaml
network:
  version: 2
  ethernet:
    enp0s3:
      nameservers:
        addresses:
          - "192.168.100.4"
        search: [insat.tn]
```

On applique l'utilisation de ce fichier avec cette commande : `sudo netplan try`



```
root@ldapclient:/home# nano /etc/netplan/00-priv_name_server.yaml
root@ldapclient:/home# netplan apply
root@ldapclient:/home# netplan apply
root@ldapclient:/home# netplan try
Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in 114 seconds
Configuration accepted.
```

On appuie sur ENTER pour accepter la nouvelle configuration.

Maintenant, on vérifie le résolveur DNS du système pour déterminer si notre configuration DNS a été appliquée :

```
sudo resolvectl status
```

Résultat:

```
Link 2 (enp0s3)
  Current Scopes: DNS
  DefaultRoute setting: yes
  LLMNR setting: yes
  MulticastDNS setting: no
  DNSOverTLS setting: no
  DNSSEC setting: no
  DNSSEC supported: no
  Current DNS Server: 8.8.8.8
  DNS Servers: 8.8.8.8
               8.8.4.4
  DNS Domain: ~.
               insat.tn
```

## 5. Test des clients

Utilisez `nslookup` pour tester si nos clients peuvent interroger nos serveurs de noms

### Forward Lookup:

Pour récupérer l'adresse IP de client avec une recherche directe :

```
nslookup client
```

### Reverse Lookup :

On interroge le serveur DNS avec l'adresse IP privée du client :

```
nslookup 192.168.56.108
```

## ▼ Partie 3 : Kerberos Authentication avec Kerberos

### ▼ Section 1 : Installez et configurez un serveur Kerberos.

#### 1.1-Installez et configurez un serveur Kerberos.

Au début, nous devons installer Kerberos en installant les packages suivants :

```
apt install krb5-kdc krb5-admin-server krb5-config
```

puis nous entrons notre DN et le nom d'hôte du serveur

Nous utilisons cette commande pour initialiser notre domaine Kerberos :

```
karam@DESKTOP-2IBK61Q:/etc$ kinit
kinit: Client's credentials have been revoked while getting initial credentials
karam@DESKTOP-2IBK61Q:/etc$ sudo krb5_newrealm
This script should be run on the master KDC/admin server to initialize
a Kerberos realm. It will ask you to type in a master key password.
This password will be used to generate a key that is stored in
/etc/krb5kdc/stash. You should try to remember this password, but it
is much more important that it be a strong password than that it be
remembered. However, if you lose the password and /etc/krb5kdc/stash,
you cannot decrypt your Kerberos database.
Loading random data
Initializing database '/var/lib/krb5kdc/principal' for realm 'ATHENA.MIT.EDU',
master key name 'K/M@ATHENA.MIT.EDU'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:

Now that your realm is set up you may wish to create an administrative
principal using the addprinc subcommand of the kadmin.local program.
Then, this principal can be added to /etc/krb5kdc/kadm5.acl so that
you can use the kadmin program on other computers. Kerberos admin
principals usually belong to a single user and end in /admin. For
example, if jruiser is a Kerberos administrator, then in addition to
the normal jruiser principal, a jruiser/admin principal should be
created.

Don't forget to set up DNS information so your clients can find your
KDC and admin servers. Doing so is documented in the administration
guide.
```

```
karam@DESKTOP-2IBK61Q:/etc$ sudo systemctl start krb5-kdc krb5-admin-server
karam@DESKTOP-2IBK61Q:/etc$ sudo systemctl enable krb5-kdc krb5-admin-server
Synchronizing state of krb5-kdc.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable krb5-kdc
Synchronizing state of krb5-admin-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable krb5-admin-server
```

## 1.2 Création des principaux

Une fois le serveur configuré, il faut ajouter des principaux (entités qui peuvent être authentifiées) et définir des politiques de mot de passe.

Nous utiliserons les commandes suivantes:

Pour créer un principal administrateur:

```
sudo kadmin.local -q "addprinc admin/admin"
```

Pour créer un utilisateur principal

```
sudo kadmin.local -q "addprinc user1"
sudo kadmin.local -q "addprinc user2"
```

Définir une politique de mot de passe:

```
sudo kadmin.local -q "addpol users"
sudo kadmin.local -q "modpol -maxlife 90d -minlength 8 u
```

Appliquer la politique de mot de passe sur les utilisateurs

```
sudo kadmin.local -q "modprinc -policy users user1"
```

code complet

```

karam@DESKTOP-2IBK61Q:/etc$ sudo kadmin.local -q "addprinc admin/admin"
Authenticating as principal root/admin@ATHENA.MIT.EDU with password.
No policy specified for admin/admin@ATHENA.MIT.EDU; defaulting to no policy
Enter password for principal "admin/admin@ATHENA.MIT.EDU":
Re-enter password for principal "admin/admin@ATHENA.MIT.EDU":
Principal "admin/admin@ATHENA.MIT.EDU" created.
karam@DESKTOP-2IBK61Q:/etc$ sudo kadmin.local -q "addprinc user1"
Authenticating as principal root/admin@ATHENA.MIT.EDU with password.
No policy specified for user1@ATHENA.MIT.EDU; defaulting to no policy
Enter password for principal "user1@ATHENA.MIT.EDU":
Re-enter password for principal "user1@ATHENA.MIT.EDU":
Principal "user1@ATHENA.MIT.EDU" created.
karam@DESKTOP-2IBK61Q:/etc$ sudo kadmin.local -q "addprinc user2"
Authenticating as principal root/admin@ATHENA.MIT.EDU with password.
No policy specified for user2@ATHENA.MIT.EDU; defaulting to no policy
Enter password for principal "user2@ATHENA.MIT.EDU":
Re-enter password for principal "user2@ATHENA.MIT.EDU":
Principal "user2@ATHENA.MIT.EDU" created.
karam@DESKTOP-2IBK61Q:/etc$
karam@DESKTOP-2IBK61Q:/etc$ sudo kadmin.local -q "addpol users"
Authenticating as principal root/admin@ATHENA.MIT.EDU with password.
karam@DESKTOP-2IBK61Q:/etc$ sudo kadmin.local -q "modpol -maxlife 90d -minlength 8 users"
Authenticating as principal root/admin@ATHENA.MIT.EDU with password.
karam@DESKTOP-2IBK61Q:/etc$ sudo kadmin.local -q "modprinc -policy users user1"
Authenticating as principal root/admin@ATHENA.MIT.EDU with password.
Principal "user1@ATHENA.MIT.EDU" modified.
karam@DESKTOP-2IBK61Q:/etc$

```

## ▼ Section 2: Authentification avec un Service Choisi

**2.1 Choisissez l'un des services (OpenLDAP, SSH, Apache, ou OpenVPN) pour implémenter l'authentification avec Kerberos.**

installez openssh

```
sudo apt install openssh-server
```

- **Configuration de OpenSSH:**

Modifier le OpenSSH configuration file `/etc/ssh/sshd_config` and `/etc/ssh/ssh_config`

```
ubuntu@ip-172-31-44-97:~$ sudo nano /etc/ssh/sshd_config
```

```

# GSSAPI options
GSSAPIAuthentication yes
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
#KerberosKeytab /etc/krb5.keytab

```

On redémarrer le service OpenSSH et applique les changements.

```
sudo systemctl restart ssh
```

Puis on ajoute la clé : cela est utilisé pour les principaux de service, où la clé est utilisée pour authentifier le service.

```
ubuntu@ip-172-31-44-97:~$ sudo ktutil
ktutil: add_entry -password -p root/admin@INSAT.TN -k 1 -e aes256-cts-hmac-sha1-96
Password for root/admin@INSAT.TN:
ktutil: wkt /etc/krb5kdc/kadm5.keytab
ktutil: q
```

```
ubuntu@ip-172-31-44-97:~$ sudo ktutil
ktutil: addent -password -p host/ec2-54-227-215-167.compute-1.amazonaws.com -k 1 -e aes256-cts-hmac-sha1-96
Password for host/ec2-54-227-215-167.compute-1.amazonaws.com@INSAT.TN:
ktutil: wkt kadm5.keytab
ktutil: q
```

```
ubuntu@ip-172-31-44-97:~$ sudo kadmin.local
Authenticating as principal root/admin@INSAT.TN with password.
kadmin.local: ktadd -k /etc/krb5kdc/kadm5.keytab kadmin/changepw
Entry for principal kadmin/changepw with kvno 3, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5kdc/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 3, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/etc/krb5kdc/kadm5.keytab.
kadmin.local: addprinc -randkey host/ec2-54-227-215-167.compute-1.amazonaws.com
No policy specified for host/ec2-54-227-215-167.compute-1.amazonaws.com@INSAT.TN; defaulting to no policy
add_principal: Principal or policy already exists while creating "host/ec2-54-227-215-167.compute-1.amazonaws.com@INSAT.TN".
kadmin.local:
```