Part 1 : OpenLDAP, SSH, Apache, OpenVPN

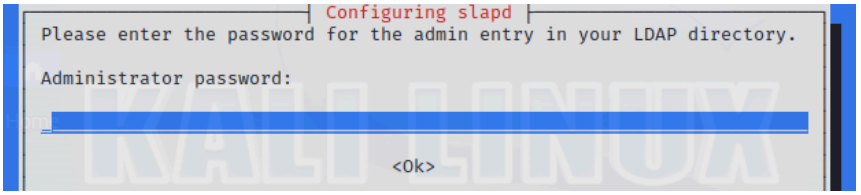
Section 1 : Configuration d'OpenLDAP

1.1 OpenLDAP Server Configuration

installing openldap server:

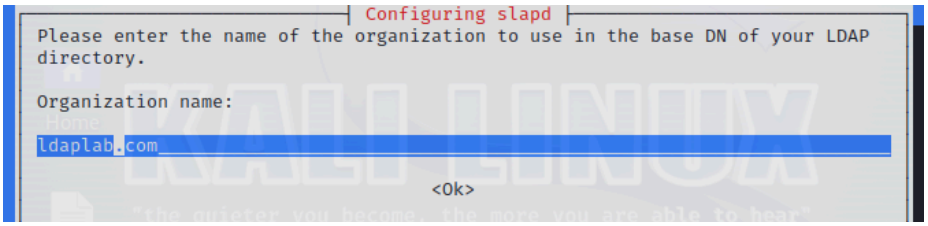
```
sudo apt-get update
sudo apt-get install slapd ldap-utils
```

configure administrator password



configure the ldap server:

```
sudo dpkg-reconfigure slapd
```



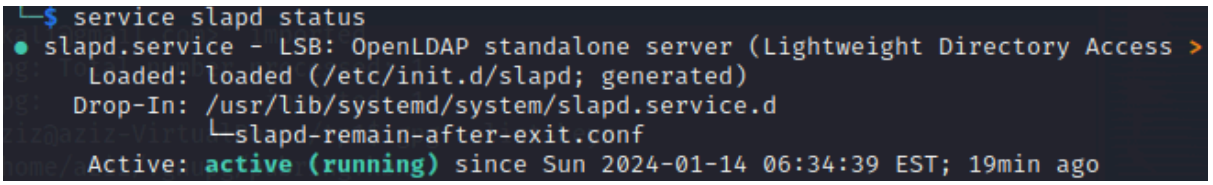
set organization name, base DN, provide the Administrator password.
remove the database when slapd is purged: select 'NO'.
select 'Yes' to remove the old database to create room for a new database.

access ldap.conf file:

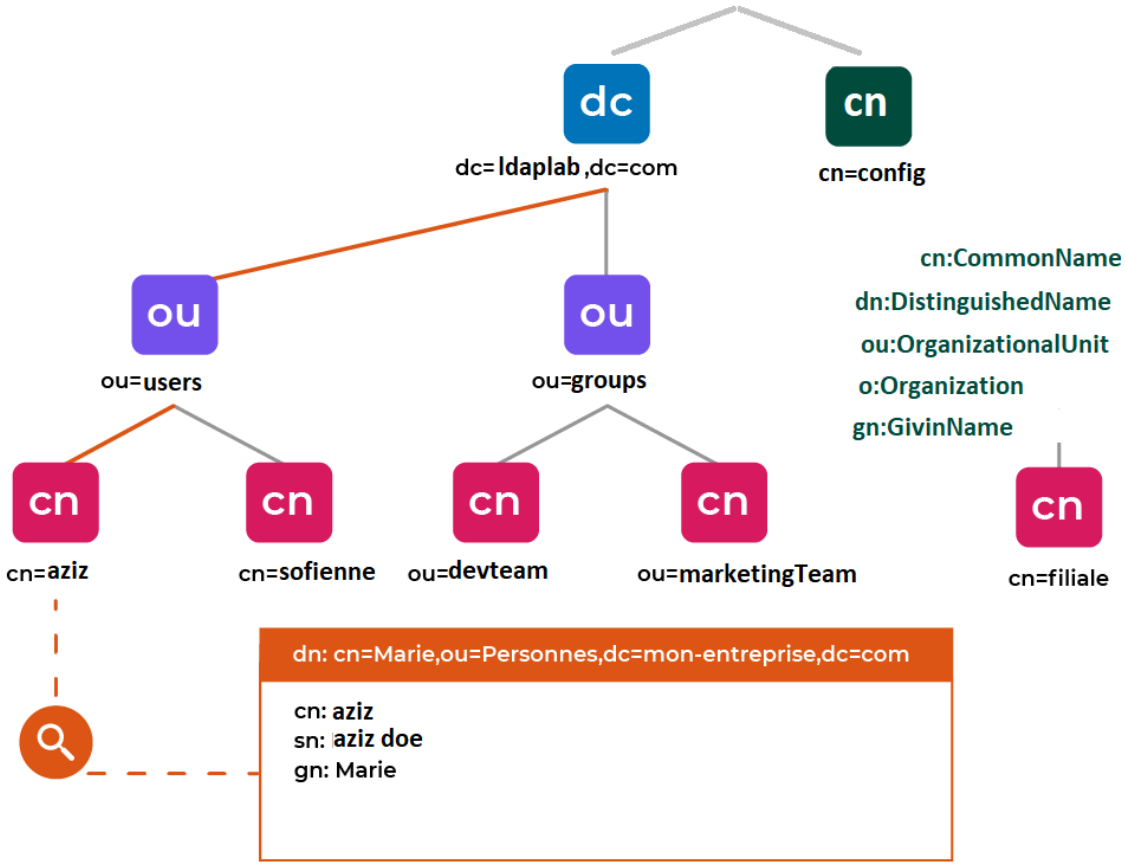
```
sudo nano /etc/ldap/ldap.conf
```

after applying changes:

```
sudo service slapd restart
sudo service slapd status
```



1.2 Directory data



for the ldap directory we added these records:
organization unit:{users, groups}
groups:{cn=devteam, marketingteam, webusers}
users:{uid=aziz, sofien, firas, heni}

create add_content.ldif: (NB:correct syntax before running command)

```
……................................................................................
dn: ou=users,dc=ldaplab,dc=com
objectClass: organizationalUnit
ou: users

dn: ou=groups,dc=ldaplab,dc=com
objectClass: organizationalUnit
ou: groups

dn: cn=devteam,ou=groups,dc=ldaplab,dc=com
objectClass: posixGroup
cn: devteam
gidNumber: 5000
memberUid:aziz

dn: cn=marketingteam,ou=groups,dc=ldaplab,dc=com
objectClass: posixGroup
cn: devteam
gidNumber: 5001
memberUid:sofien
```

```
dn: cn=webusers,ou=groups,dc=ldaplab,dc=com
objectClass: top
objectClass: groupOfNames
cn: webusers
description: web Users
member: uid=sofien,ou=users,dc=ldaplab,dc=com
member: uid=firas,ou=users,dc=ldaplab,dc=com

dn: uid=aziz,ou= users,dc=ldaplab,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: aziz
sn: doe
givenName: aziz
cn: aziz doe
displayName: aziz doe
uidNumber: 10000
gidNumber: 5000
userPassword: {CRYPT}x
gecos: aziz doe
loginShell: /bin/bash
homeDirectory: /home/aziz

dn: uid=sofien,ou= users,dc=ldaplab,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: sofien
sn: da
givenName: sofien
cn: sofien da
displayName: sofien da
uidNumber: 10001
gidNumber: 5001
userPassword: {CRYPT}x
gecos: sofien da
loginShell: /bin/bash
homeDirectory: /home/sofien

dn: uid=firas,ou= users,dc=ldaplab,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: firas
sn: da
givenName: firas
cn: firas da
displayName: firas da
uidNumber: 10002
gidNumber: 5001
userPassword: {CRYPT}x
gecos: firas da
loginShell: /bin/bash
homeDirectory: /home/firas
```

……

for every user we assigned a given name, homeDirectory, displayName…

let's add now the content:

```
ldapadd -x -D cn=admin,dc=ldaplab,dc=com -W -f add_content.txt
```

```
┌──(aziz㉿kali)-[~/sec_project]
└─$ ldapadd -x -D cn=admin,dc=ldaplab,dc=com -W -f add_content.txt
Enter LDAP Password:
adding new entry "uid=aziz,ou= users,dc=ldaplab,dc=com"

adding new entry "uid=sofien,ou= users,dc=ldaplab,dc=com"
```

make ldap request to check directory content:

```
ldapsearch -x -LLL -b dc=ldaplab,dc=com dn:
```

```
┌──(aziz㉿kali)-[~/sec_project]
└─$ ldapsearch -x -LLL -b dc=ldaplab,dc=com dn:
dn: dc=ldaplab,dc=com

dn: ou=users,dc=ldaplab,dc=com

dn: ou=groups,dc=ldaplab,dc=com

dn: cn=devteam,ou=groups,dc=ldaplab,dc=com

dn: cn=marketingteam,ou=groups,dc=ldaplab,dc=com

dn: uid=aziz,ou=users,dc=ldaplab,dc=com

dn: uid=sofien,ou=users,dc=ldaplab,dc=com
```

change users password:

```
ldappasswd -H ldapi:/// -x -D cn=admin,dc=ldaplab,dc=com -W -S
uid=aziz,ou=users,dc=ldaplab,dc=com
```

```
┌──(aziz㉿kali)-[~/sec_project]
└─$ ldappasswd -H ldapi:/// -x -D cn=admin,dc=ldaplab,dc=com -W -S uid=aziz,o
u=users,dc=ldaplab,dc=com
New password:
Re-enter new password:
Enter LDAP Password:
```

add x509 Certificate for each user:

start by generating x509 certificates: (self signed certificates)

```
┌──(aziz㉿kali)-[~/sec_project]
└─$ openssl x509 -req -in aziz_csr.pem -signkey aziz_key.pem -out aziz_cert.p
em

Certificate request self-signature ok
subject=C = tn, ST = Tunis, L = tunis, O = insat, OU = gl, CN = aziz, emailAd
dress = aziz@gmail.com
┌──(aziz㉿kali)-[~/sec_project]
└─$ openssl x509 -req -in sofien_csr.pem -signkey sofien_key.pem -out sofien_
cert.pem
```

the certificate must be published into the LDAP server as a binary piece of data.
we encode the certificate in BASE64 and parse it into ldif modifying file:

add_certif.ldif:

```
dn: uid=aziz,ou=users,dc=ldaplab,dc=com
changetype: modify
add: usercertificate
usercertificate;binary:: MIIC2TCCAkKgAwIBAgIBAD…
```

```
ldapmodify -x -D "cn=admin,dc=ldaplab,dc=com" -W -f add_certif.ldif
```

check:

```
┌──(aziz㉿kali)-[~/sec_project]
└─$ ldapsearch -x -LLL -b dc=ldaplab,dc=com '(uid=aziz)' userCertificate
dn: uid=aziz,ou=users,dc=ldaplab,dc=com
userCertificate;binary:: MIIDdzCCAl8CFGbmAd0ChHCs+jfbckZdxiXXyvXDMA0GCSqGSIb3
D
 QEBCwUAMHgxCzAJBgNVBAYTAnRuMQ4wDAYDVQQIDAVUdW5pczEOMAwGA1UEBwwFdHVuaXMxDjAMB
```

1.3 Ensure that users can successfully authenticate to the OpenLDAP server

identifying with admin:

```
ldapmodify -x -D "cn=admin,dc=ldaplab,dc=com" -W -f
add_user_to_devteam.ldif
```

with simple user:

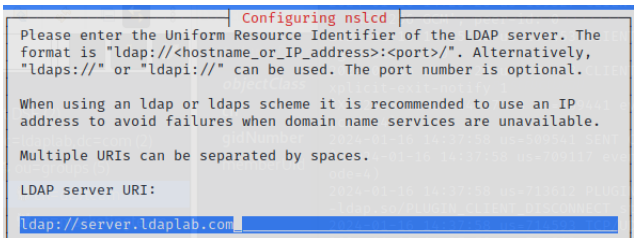
```
ldapmodify -x -D "uid=firas,dc=ldaplab,dc=com" -W -f
add_user_to_devteam.ldif
```

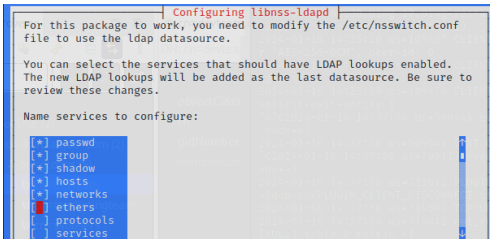
```
┌──(aziz㉿kali)-[~/sec_project]
└─$ sudo ldapsearch -x -D uid=aziz,ou=users,dc=ldaplab,dc=com -W -b dc=ldapla
b,dc=com

Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=ldaplab,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# ldaplab.com
dn: dc=ldaplab,dc=com
objectClass: top
```

distant ldap client:

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

a nslcd configuration will prompt





we can change that after that manually:

`nano /etc/nslcd.conf` `nano /etc/nsswitch.conf`

now we can try login with ldap users

```
aziz@aziz-VirtualBox:~/Desktop$ su -l firas
Password:
firas@aziz-VirtualBox:~$
```

make a search form client

```
sofien@aziz-VirtualBox:/home/aziz$ ldapsearch -x -H ldap://192.168.56.101 -b "dc
=ldaplab,dc=com"
# extended LDIF
#
# LDAPv3
# base <dc=ldaplab,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# ldaplab.com
dn: dc=ldaplab,dc=com
```

1.4 LDAPs (ldap secure)

Pour s'authentifier sur un serveur OpenLDAP, on va

generate certificates:

autorité de certification (CA):

```
openssl genrsa -out CA.key 2048
```

```
openssl req -x509 -nodes -days 100000 -key insat.key  -out cert_CA
```

```
┌──(aziz㉿server)-[~/sec_project]
└─$ openssl genrsa -out key_CA.pem 2048

┌──(aziz㉿server)-[~/sec_project]
└─$ openssl req -x509 -nodes -days 100000 -key key_CA.pem  -out cert_CA.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

NB: We are placing the CA certificate in the directory /usr/local/share/ca-certificates, where update-ca-certificates will retrieve trusted local CAs. If you want to include CAs from /usr/share/ca-certificates, you need to execute dpkg-reconfigure ca-certificates.

```
└─$ sudo mv cert_CA.pem /usr/local/share/ca-certificates/cert_CA.crt
```

```
┌──(aziz㉿server)-[~/sec_project]
└─$ ls /usr/local/share/ca-certificates/
cert_CA.crt

┌──(aziz㉿server)-[~/sec_project]
└─$ sudo dpkg-reconfigure ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one cer
tificate or CRL
1 added, 0 removed; done.
Processing triggers for ca-certificates (20230311)
```

server private key:

```
openssl genrsa -out key_server.pem 2048
```

server CSR:

```
openssl req  -key key_server.pem  -out csr_server.pem -new
```

Sign the CSR with the self-signed certificate's private key

```
openssl x509 -req -in csr_server.pem -out cert_server.pem -CA cert_CA.pem  -CAkey
key_CA.pem -CAreateserial -days 365

mv cert_server.pem cert_server.crt
```

```
┌──(aziz㉿server)-[~/sec_project]
└─$ openssl x509 -req -in csr_server.pem -out cert_server.pem -CA cert_CA.pem  -C
Akey key_CA.pem -CAcreateserial -days 365
Certificate request self-signature ok
subject=C = tn, ST = tunis, L = insat, O = ldap, OU = ldapserver, CN = ldapserver
, emailAddress = ldapserver@gmail.com

┌──(aziz㉿server)-[~/sec_project]
└─$ cat cert_server.pem
-----BEGIN CERTIFICATE-----
MIIDmTCCAoECFGA67BW9mYfbGOBT0Htjhu1OMYLMMA0GCSqGSIb3DQEBCwUAMIGF
```

place certifs:

```
sudo cp key_server.pem /etc/ldap/sasl2/

sudo cp cert_server.crt /etc/ldap/sasl2/

sudo cp /etc/ssl/certs/ca-certificates.crt /etc/ldap/sasl2/
```

```
┌──(aziz㉿server)-[~/sec_project]
└─$ ll /etc/ldap/sasl2
total 220
-rw-r--r-- 1 root root 215201 Jan 14 12:18 ca-certificates.crt
-rw-r--r-- 1 root root   1310 Jan 14 12:18 cert_server.crt
-rw------- 1 root root   1704 Jan 14 12:17 key_server.pem
```

```
└─$ sudo mv /etc/ldap/sasl2/cert_server.crt /etc/ldap/sasl2/server.ldaplab.co
m.crt

┌──(aziz㉿server)-[~/sec_project]
└─$ sudo mv /etc/ldap/sasl2/key_server.pem /etc/ldap/sasl2/server.ldaplab.com
.key
```

`sudo chown -R openldap:openldap /etc/ldap/sasl2/`

```
└─$ ll /etc/ldap/sasl2
total 220
-rw-r--r-- 1 openldap openldap 215201 Jan 14 12:18 ca-certificates.crt
-rw-r--r-- 1 openldap openldap   1310 Jan 14 12:18 server.ldaplab.com.crt
-rw------- 1 openldap openldap   1704 Jan 14 12:17 server.ldaplab.com.key
```

certinfo.ldif

```
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/sasl2/ca-certificates.crt
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/sasl2/server.ldaplab.com.crt
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/sasl2/server.ldaplab.com.key
```

`sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif`

`nano /etc/default/slapd` add ldaps:///

```
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:///
SLAPD_SERVICES="ldap:/// ldapi:/// ldaps:///"
```

`nano /etc/default/ldap.conf`

```
# TLS certificates (needed for GnuTLS)
#TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
TLS_CACERT      /etc/ldap/sasl2/ca-certificates.crt
TLS_REQCERT     allow
```

access via ldaps: it works 👍

```
└─$ ldapsearch -x -H ldap://server.ldaplab.com
# extended LDIF
#
# LDAPv3
# base <dc=ldaplab,dc=com> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# ldaplab.com
dn: dc=ldaplab,dc=com
objectClass: top
```

ldaps advantages:

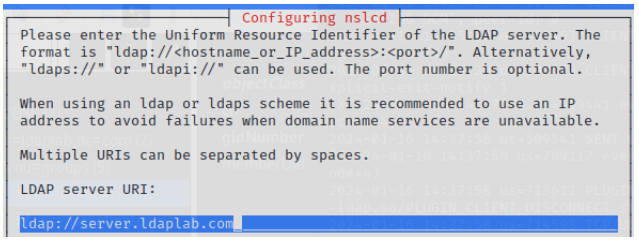
LDAPS secures LDAP communications through encrypted SSL/TLS connections on port 636. Advantages include data encryption, protection against passive listening, server authentication, strong user authentication, compliance with security standards, LDAP application compatibility, security in untrusted networks, and certificate-based access control. In summary, LDAPS ensures confidentiality, authenticity, and data integrity, though StartTLS is increasingly favored in some cases.

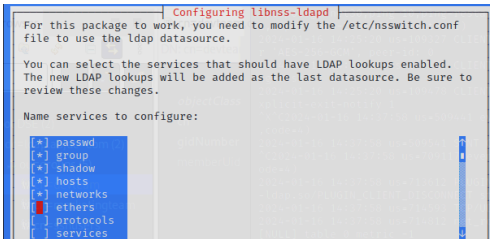
Section 2 : SSH Authentication

2.1 Activation de l'Authentification SSH via OpenLDAP

`sudo apt-get install openssh-server libpam-ldap`

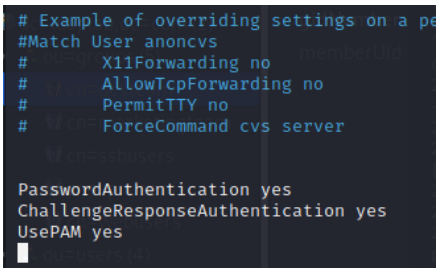
a nslcd configuration will prompt



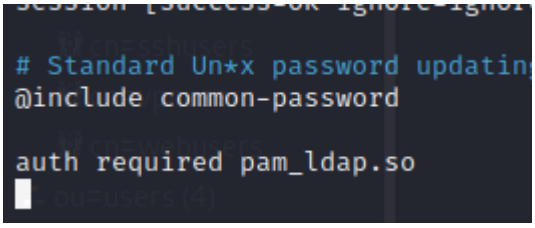


`nano /etc/nslcd.conf` `nano /etc/nsswitch.conf`

`sudo nano /etc/ssh/sshd_config` add the following



`sudo nano /etc/pam.d/sshd`



2.2 Restrict SSH access to a group of users

assign a group to users and only allow that group to connect via SSH.

`sudo nano /etc/security/access.conf` add GROUP_ldapgroup

```
#
# All other users should be denied to g
#-:ALL:ALL
-:ALL EXCEPT GROUP_marketingteam:ALL
```

`sudo service ssh restart`

`sudo service nslcd restart`

let's test:

aziz:devteam not allow to authenticate

sofien:marketingteam yes

```
aziz@aziz-VirtualBox:~/sec_project$ ssh aziz@server
aziz@server's password:
Permission denied, please try again.
aziz@server's password:

aziz@aziz-VirtualBox:~/sec_project$ ssh sofien@server
sofien@server's password:
Linux server 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023
06-29) x86_64

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 16 15:13:45 2024 from 192.168.56.102
Could not chdir to home directory /home/sofien: No such file or directory
sofien@server:/$
```

Section 3 : Integration of Apache

3.1 Enable SSH authentication via OpenLDAP

1) `sudo apt-get install apache2 libapache2-mod-ldap-userdir`

This command sets up Apache on your system and includes the `mod_ldap_userdir` module, enabling user-specific web directories and allowing authentication against an LDAP directory. After installation, you may need to configure Apache and the LDAP module based on your specific requirements.

2) Enable LDAP module in the apache

```
sudo a2enmod ldap
```

This command enables the Apache module `mod_ldap`. The `mod_ldap` module provides support for basic LDAP authentication in Apache. When enabled, it allows Apache to authenticate users against an LDAP directory.

```
sudo a2enmod authnz_ldap
```

This command enables the Apache module `mod_authnz_ldap`. The `mod_authnz_ldap` module extends LDAP support by providing additional features for authentication and authorization. It includes directives that allow you to specify LDAP-related authentication and authorization configurations directly in Apache's configuration files.

3) Add this code to /etc/apache2/sites-available/000-default.conf file

```
<Directory /var/www/html>
AuthType Basic AuthName "LDAP Authentication"
AuthBasicProvider ldap
AuthLDAPURL "ldap://<LDAP_SERVER_IP>/dc=ldaplab,dc=com?uid"
Require valid-user
 </Directory>
```

NB: AuthzLDAPAuthoritative Off: If set to Off, it means that other authorization modules are authoritative, and LDAP decisions are only considered if other modules do not make a definitive decision.

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostna
        # the server uses to identify itself. This is used when cr
        # redirection URLs. In the context of virtual hosts, the S
        # specifies what hostname must appear in the request's Hos
        # match this virtual host. For the default virtual host (t
        # value is not decisive as it is used as a last resort hos
        # However, you must set it for any further virtual host ex
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
<Directory /var/www/html>
    AuthType Basic
    AuthName "LDAP Authentication"
    AuthBasicProvider ldap
    AuthLDAPURL "ldap://server.ldaplab.com/dc=ldaplab,dc=com?uid"
    Require valid-user
</Directory>
```

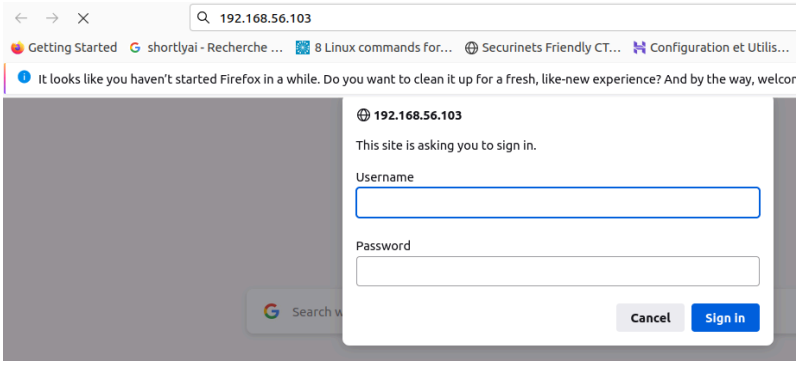
3.2 Restrict SSH access to users from the appropriate group in OpenLDAP

Constrain SSH access exclusively to users belonging to the designated group within OpenLDAP. Subsequently, replace the directive 'required valid-user' with 'Require ldap-group cn=webusers,ou=groups,dc=ldaplab,dc=com' to define the permitted group.

```
<Directory /var/www/html>
    AuthType Basic
    AuthName "LDAP Authentication"
    AuthBasicProvider ldap
    AuthLDAPURL "ldap://server.ldaplab.com/dc=ldaplab,dc=com?uid"
    Require ldap-group cn=webusers,ou=groups,dc=ldaplab,dc=com
</Directory>
```

3.3 Test for an authorized user and an unauthorized user for SSH

Testing: trying to access apache web page from a distant machine via port 80:



Section4: Setting up OpenVPN

4.1 Install and configure OpenVPN to use OpenLDAP authentication

1) `sudo apt install openvpn easy-rsa`

You will have OpenVPN installed on your system, and Easy-RSA will be available to assist in the creation and management of the cryptographic keys and certificates needed for secure communication within the OpenVPN setup.

2)Now we generate easy-rsa directory that contains the necessary scripts:

`make-cadir easy-rsa`

```
└─$ make-cadir easy-rsa

┌──(aziz㉿server)-[~/sec_project/vpn-ldap]
└─$ cd easy-rsa

┌──(aziz㉿server)-[~/sec_project/vpn-ldap/easy-rsa]
└─$ ls
easyrsa  openssl-easyrsa.cnf  vars  x509-types
```

`./easyrsa init-pki`

```
└─$ ./easyrsa init-pki

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

Your newly created PKI dir is:
* /home/aziz/sec_project/vpn-ldap/easy-rsa/pki

Using Easy-RSA configuration:
* /home/aziz/sec_project/vpn-ldap/easy-rsa/vars
```

`./easyrsa build-ca  >> ca.crt` (save the ca_key)

`./easyrsa gen-dh` (generate key Diffie-Hellman (DH): dh.pem)

`./easyrsa build-server-full SERVER nopass`

`./easyrsa build-client-full CLIENT nopass`

`openvpn --genkey secret ta.key` (encrypt the tunnel)

for routing: nano `/etc/sysctl.conf` and uncomment: `net.ipv4.ip_forward=1`

then `sysctl -p /etc/sysctl.conf`

3)Now we are going to put the generated information in their place under openvpn client and server, we can transfer files to client by scp:

(server side)`cp ta.key pki/dh.pem pki/ca.crt pki/issued/SERVER.crt pki/private/SERVER.key /etc/openvpn`

(from server to client) `scp ta.key pki/ca.crt pki/issued/CLIENT.crt pki/private/CLIENT.key`

`whoami@(hostname -I):/home/aziz`

(client side) cp ta.key ca.crt CLIENT.crt CLIENT.key /etc/openvpn

CONFIG FILES:

(server side) `nano /etc/openvpn/server.conf:`

`# listen on? (optional)`

`;local a.b.c.d`

`# open up this port on your firewall.`

```
port 1194
# TCP or UDP server?
;proto tcp
proto udp
;dev tap
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/SERVER.crt
key /etc/openvpn/SERVER.key  # This file should be kept secret
dh /etc/openvpn/dh.pem
server 10.6.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
keepalive 10 120
tls-auth /etc/openvpn/ta.key 0 # This file is secret
cipher AES-256-CBC
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
verb 3
explicit-exit-notify 1
```

`sudo openvpn /etc/openvpn/server.conf` (this command will open the tun1)

`sudo systemctl restart openvpn@server`

we can check:

```
└─$ hostname -I
10.0.2.15 192.168.56.103 10.6.0.1
```

CONFIG FILES:

(client side)

`nano /etc/openvpn/client.conf:`

```
client
dev tun
proto udp
```

```
# resplace server.openvpn.com with your server ip
remote server.openvpn.com 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/CLIENT.crt
key /etc/openvpn/CLIENT.key
remote-cert-tls server
# If a tls-auth key is used on the server
# then every client must also have the key.
tls-auth /etc/openvpn/ta.key 1
cipher AES-256-CBC
verb 3
```

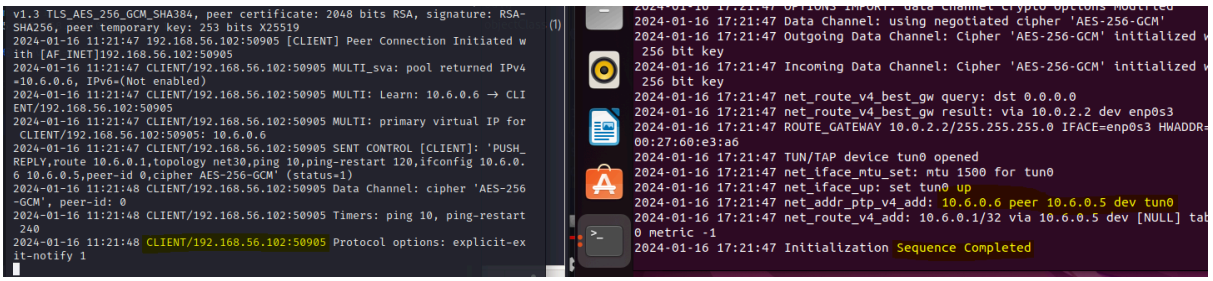
4)It is crucial not to overlook the addition of the host address

```
  GNU nano 6.2
127.0.0.1        localhost
127.0.1.1        aziz-VirtualBox
192.168.56.103   server.ldaplab.com server server.openvpn.com
```

5)launch the OpenVPN client with the specified configuration file

```
sudo openvpn /etc/openvpn/client.conf
```

6)successful connection



4.2 Successfully test the VPN connection using the information from OpenLDAP

1)`sudo apt install openvpn-auth-ldap`

2) Once openvpn-auth-ldap package is installed, the required module will be deployed at

```
/usr/lib/openvpn/openvpn-auth-ldap.so
mkdir /etc/openvpn/auth
cp /usr/share/doc/openvpn-auth-ldap/examples/auth-ldap.conf
/etc/openvpn/auth/
nano /etc/openvpn/auth/auth-ldap.conf:
<LDAP>
      # LDAP server URL
      URL         ldap://server.ldaplab.com
      # Network timeout (in seconds)
      Timeout          15
      # Enable Start TLS
      TLSEnable   no
      # Follow LDAP Referrals (anonymously)
      FollowReferrals no
</LDAP>
<Authentication>
    # Base DN
    BaseDN      "ou=users,dc=ldaplab,dc=com"
    # User Search Filter
SearchFilter"(&(uid=%u)(accountStatus=active)(memberOf=cn=vpnusers,ou=g
roups,dc=ldaplab,dc=com))"
    # Require Group Membership
    RequireGroup    true
</Authentication>
```

3) Now we add the server.conf file

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

4) Now for the client

we add this to the client.conf:

```
auth-user-pass
```

5) `systemctl restart openvpn-server@server`

An error may occur in this case it is appropriate to terminate the OpenVPN process and restart it.

result: vpnusers

```
  GNU nano 7.2                    vpnusersgroup.ldif
dn: cn=vpnusers,ou=groups,dc=ldaplab,dc=com
objectClass: top
objectClass: groupOfNames
cn: vpnusers
description: vpn users
member: uid=firas,ou=users,dc=ldaplab,dc=com
member: uid=aziz,ou=users,dc=ldaplab,dc=com
```

6)Non-authorized user:(it doesn't work 👎)

```
2024-01-16 12:29:42 192.168.56.102:38039 TLS: move_session: dest=TM_ACTIVE sr
c=TM_INITIAL reinit_src=1
2024-01-16 12:29:42 192.168.56.102:38039 TLS: tls_multi_process: initial untr
usted session promoted to semi-trusted
2024-01-16 12:29:42 192.168.56.102:38039 Delayed exit in 5 seconds
2024-01-16 12:29:42 192.168.56.102:38039 SENT CONTROL [UNDEF]: 'AUTH_FAILED'
(status=1)
2024-01-16 12:29:42 192.168.56.102:38039 SENT CONTROL [CLIENT]: 'AUTH_FAILED'
 (status=1)
2024-01-16 12:29:42 192.168.56.102:38039 Control Channel: TLSv1.3, cipher TLS
v1.3 TLS_AES_256_GCM_SHA384, peer certificate: 2048 bits RSA, signature: RSA-
SHA256, peer temporary key: 253 bits X25519
2024-01-16 12:29:42 192.168.56.102:38039 [CLIENT] Peer Connection Initiated w
ith [AF_INET]192.168.56.102:38039
2024-01-16 12:29:44 read UDPv4 [ECONNREFUSED]: Connection refused (fd=7,code=
111)
2024-01-16 12:29:47 192.168.56.102:38039 SIGTERM[soft,delayed-exit] received,
```

```
2024-01-16 18:28:37 library versions: OpenSSL 3.0.2 15
Enter Auth Username: sofien
🔐 Enter Auth Password: ****
2024-01-16 18:29:41 Outgoing Control Channel Authentica
ge hash 'SHA1' for HMAC authentication
2024-01-16 18:29:41 Incoming Control Channel Authentica
ge hash 'SHA1' for HMAC authentication
2024-01-16 18:29:41 TCP/UDP: Preserving recently used r
92.168.56.103:1194
2024-01-16 18:29:41 Socket Buffers: R=[212992->212992]
2024-01-16 18:29:41 UDP link local: (not bound)
2024-01-16 18:29:41 UDP link remote: [AF_INET]192.168.5
2024-01-16 18:29:41 TLS: Initial packet from [AF_INET]1
4bbd735 104622d1
2024-01-16 18:29:41 WARNING: this configuration may cac
```

Authorized user:(it works 👍)

```
ate or username to concurrently connect.
2024-01-16 14:25:19 us=18835 MULTI_sva: pool returned IPv4=10.5.0.6, IPv6=(No
t enabled)
2024-01-16 14:25:19 us=19936 PLUGIN_CALL: POST /usr/lib/openvpn/openvpn-auth-
ldap.so/PLUGIN_CLIENT_CONNECT status=0
2024-01-16 14:25:19 us=20132 OPTIONS IMPORT: reading client specific options
from: /tmp/openvpn_cc_225b756e1e621a6d485954625a6cc067.tmp
2024-01-16 14:25:19 us=20751 MULTI: Learn: 10.5.0.6 → CLIENT/192.168.56.102:
57714
2024-01-16 14:25:19 us=20795 MULTI: primary virtual IP for CLIENT/192.168.56.
102:57714: 10.5.0.6
2024-01-16 14:25:19 us=21093 Data Channel MTU parms [ mss_fix:1400 max_frag:0
 tun_mtu:1500 tun_max_mtu:1600 headroom:136 payload:1768 tailroom:562 ET:0 ]
2024-01-16 14:25:19 us=21406 Outgoing Data Channel: Cipher 'AES-256-GCM' init
ialized with 256 bit key
2024-01-16 14:25:19 us=21433 Incoming Data Channel: Cipher 'AES-256-GCM' init
ialized with 256 bit key
2024-01-16 14:25:19 us=21498 SENT CONTROL [CLIENT]: 'PUSH_REPLY,route 10.5.0.
1,topology net30,ping 10,ping-restart 120,ifconfig 10.5.0.6 10.5.0.5,peer-id
1,cipher AES-256-GCM' (status=1)
2024-01-16 14:25:20 us=109327 CLIENT/192.168.56.102:57714 Data Channel: ciphe
r 'AES-256-GCM', peer-id: 0
2024-01-16 14:25:20 us=109453 CLIENT/192.168.56.102:57714 Timers: ping 10, pi
ng-restart 240
2024-01-16 14:25:20 us=109478 CLIENT/192.168.56.102:57714 Protocol options: e
xplicit-exit-notify 1
```

```
2024-01-16 20:25:18 OPTIONS IMPORT: timers and/or timeouts modified
2024-01-16 20:25:18 OPTIONS IMPORT: --ifconfig/up options modified
2024-01-16 20:25:18 OPTIONS IMPORT: route options modified
2024-01-16 20:25:18 OPTIONS IMPORT: peer-id set
2024-01-16 20:25:18 OPTIONS IMPORT: adjusting link_mtu to 1624
2024-01-16 20:25:18 OPTIONS IMPORT: data channel crypto options modified
2024-01-16 20:25:18 Data Channel: using negotiated cipher 'AES-256-GCM'
2024-01-16 20:25:18 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized wi
 256 bit key
2024-01-16 20:25:18 Incoming Data Channel: Cipher 'AES-256-GCM' initialized wi
 256 bit key
2024-01-16 20:25:18 net_route_v4_best_gw query: dst 0.0.0.0
2024-01-16 20:25:18 net_route_v4_best_gw result: via 10.0.2.2 dev enp0s3
2024-01-16 20:25:18 ROUTE_GATEWAY 10.0.2.2/255.255.255.0 IFACE=enp0s3 HWADDR=0
00:27:00:e3:a6
2024-01-16 20:25:18 TUN/TAP device tun0 opened
2024-01-16 20:25:18 net_iface_mtu_set: mtu 1500 for tun0
2024-01-16 20:25:18 net_iface_up: set tun0 up
2024-01-16 20:25:18 net_addr_ptp_v4_add: 10.5.0.6 peer 10.5.0.5 dev tun0
2024-01-16 20:25:18 net_route_v4_add: 10.5.0.1/32 via 10.5.0.5 dev [NULL] table
0 metric -1
2024-01-16 20:25:18 Initialization Sequence Completed
```

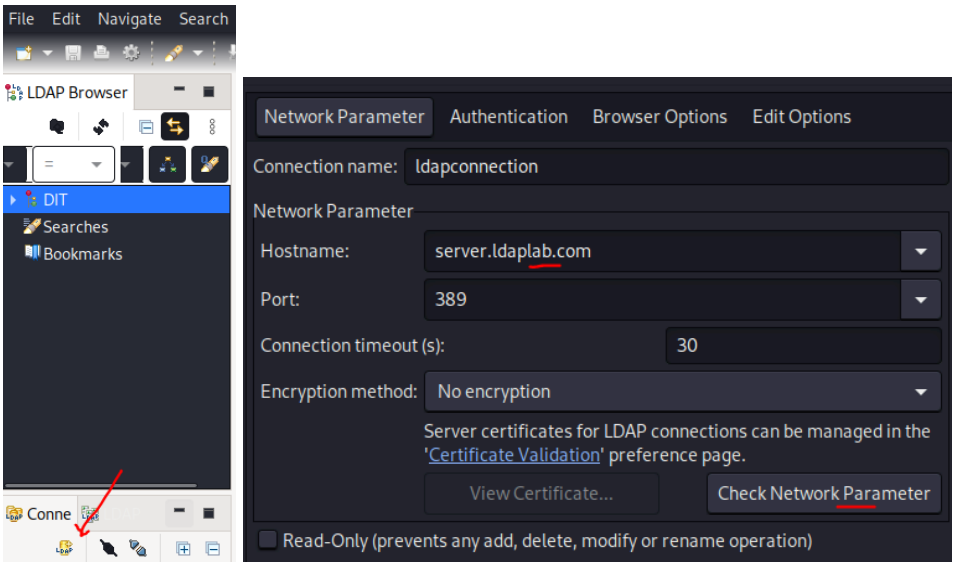
Apache directory : ----------

configure openldap via apache directory

install ldap apache directory on linux: https://www.youtube.com/watch?v=R0ocFk-pLQ8

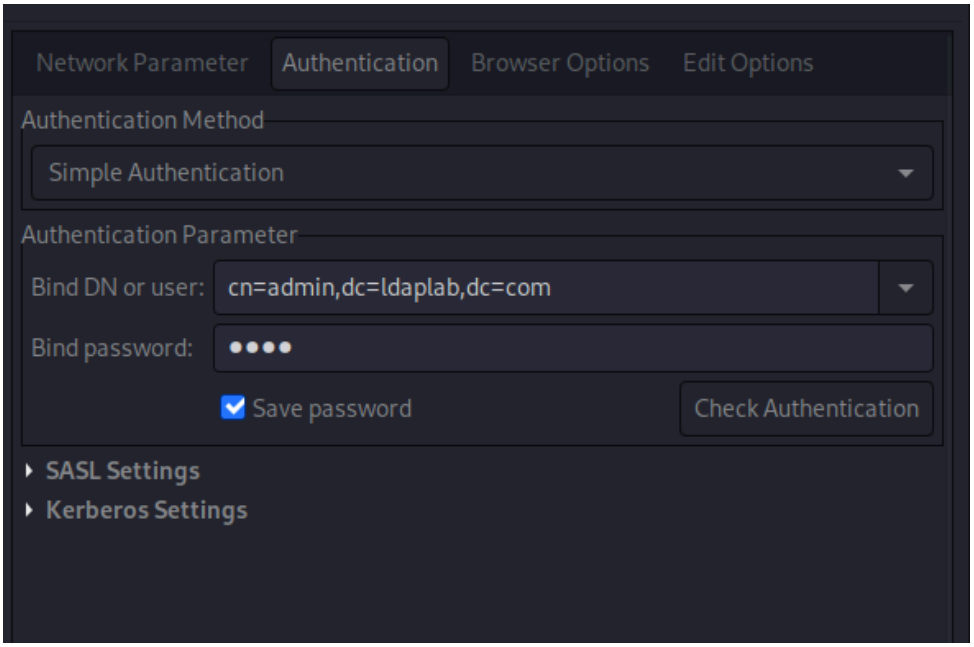
https://directory.apache.org/studio/download/download-linux.html

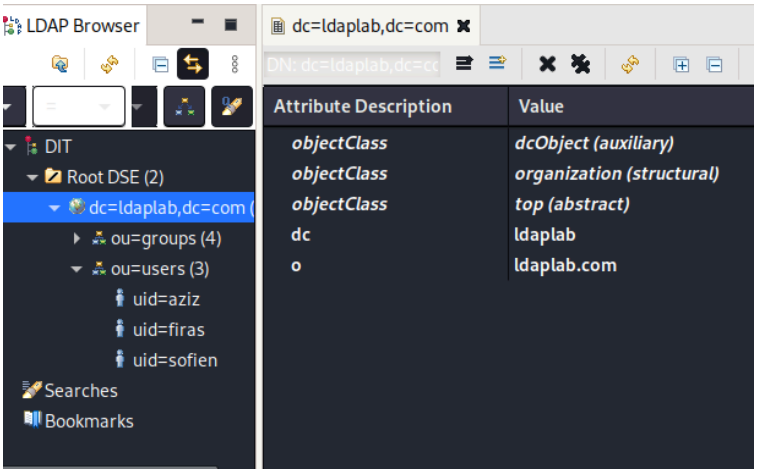
new connection>



authentication>

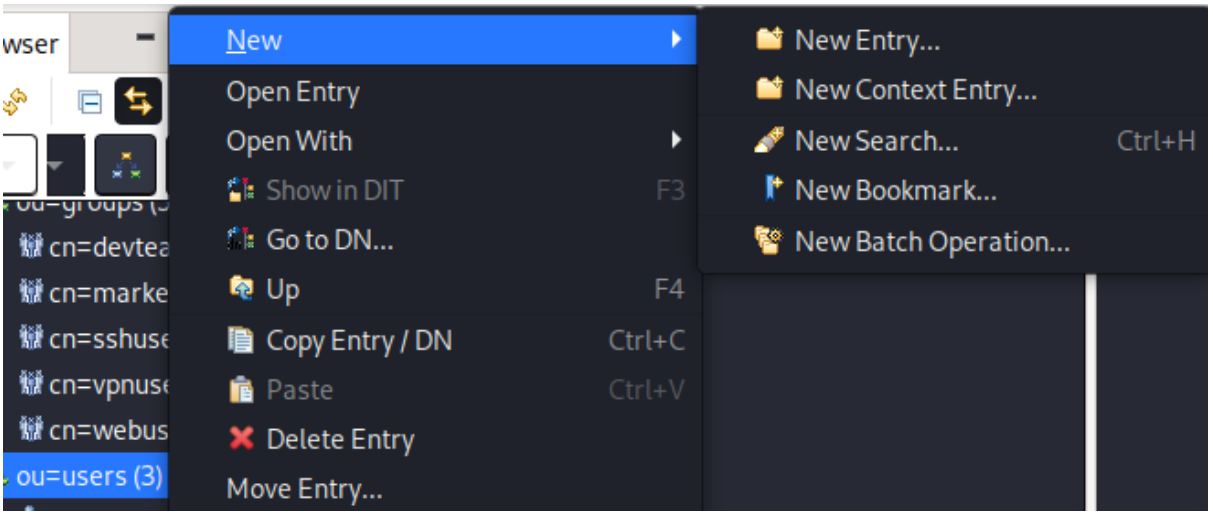
to modify the server auth is needed:



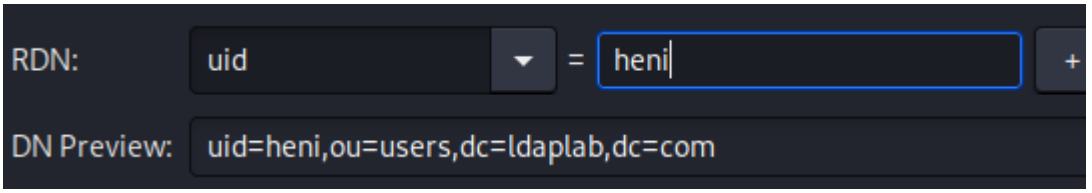


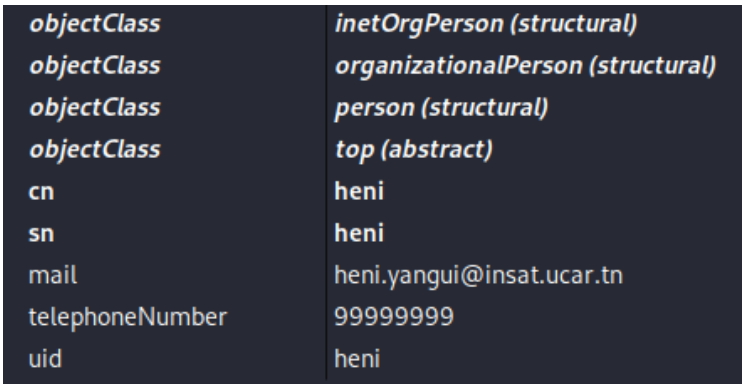
adding a new entry with apache directory:

ou:users> New >New Entry

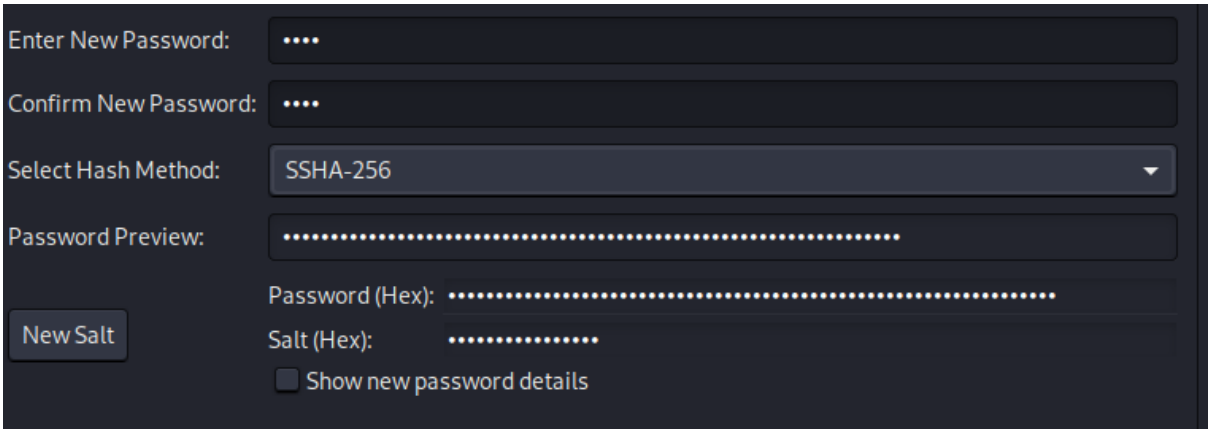


create entry from scratch > add objectClass(exp:inetOrgPerson)> add RDN(relative distinguished name)(exp:uid)> add properties





to add password : new attribute> userpassword (uid:heni, password:4444) (uid=firas, password=5555)



end apache directory: ----------

Part 2 : Dns

Section 1 : Configuring DNS Bind

We will use a DNS Server for our services. This allows us to use easy-to-remember domain names instead of remembering and updating specific IP addresses when they change.

1.1 Configurez un serveur DNS (Bind) sur une machine distincte.

`sudo apt-get install bind9`

1.2 Ajoutez les enregistrements DNS nécessaires pour les serveurs OpenLDAP, Apache, et OpenVPN.

`sudo nano /etc/bind/named.conf` add DNS zone:

```
zone "security.tn" {
   type master;
   file "/etc/bind/db.security.tn";
```

```
};
```

Create the zone files for each service and add DNS records.

```
; /etc/bind/db.security.tn
$TTL    604800
@       IN      SOA     ns1.security.tn. admin.security.tn. (
                        3         ; Serial
                   604800         ; Refresh
                    86400         ; Retry
                  2419200         ; Expire
                   604800 )       ; Negative Cache TTL
; Name servers
@       IN      NS      ns1.security.tn.
; IP addresses for name servers
ns1     IN      A       172.20.128.10
; Sample hosts
openldap        IN      A       172.20.128.2
ssh             IN      A       172.20.128.4
apache          IN      A       172.20.128.6
openvpn         IN      A       172.20.128.8
kdc             IN      A       172.20.128.12
```

```
sudo service bind9 restart
```

Section 2 : validation and test

2.1 Testez la résolution DNS pour chacun des services configurés.

For each service machine, `nano /etc/resolv.conf` and add `nameserver <dns_server_ip>` . Now every service machine is using our dns server. We restart the services to check whether they are working or not.

2.2 Assurez-vous que les noms de domaine associés aux services sont correctement résolus.

```
nslookup openldap.security.tn
nslookup openvpn.security.tn
nslookup ssh.security.tn
nslookup apache.security.tn
```

Part 3 : kerberos

General idea:

We will use two machines to authenticate a client to an SSH service using a TGT ticket given by a KDC.

Section 1:

1- machine's set up:

1.1- Since Kerberos relies on timestamps to issue and verify tickets, we should make sure our machines are synchronized . We can do such a thing using ntp protocol, we will skip this phase since the two machines are correctly synchronized.

1.2- In each machine match different ips to their sub domain name in /etc/hosts :

```
  GNU nano 6.2                    /etc/hosts *
127.0.0.1       localhost
127.0.1.1       ubunto1.myguest.virtualbox.org  ubunto1
192.168.56.101 kdc.insat.tn kdc
192.168.56.102 client.insat.tn client
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
  GNU nano 7.2                    /etc/hosts *
127.0.0.1       localhost
127.0.1.1       kali
192.168.56.101 kdc.insat.tn kdc
192.168.56.102 client.insat.tn client
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
```

To test if everything is working properly try this command on each sub-domain:

```
host kdc.insat.tn
```

1.3- Configure KDC:

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

```
Configuring Kerberos Authentication

When users attempt to use Kerberos and specify a principal or user name
without specifying what administrative Kerberos realm that principal
belongs to, the system appends the default realm.  The default realm may
also be used as the realm of a Kerberos service running on the local
machine.  Often, the default realm is the uppercase version of the local
DNS domain.

Default Kerberos version 5 realm:

INSAT.TN

<Ok>
```

```
Configuring Kerberos Authentication
Enter the hostnames of Kerberos servers in the INSAT.TN Kerberos realm
separated by spaces.

Kerberos servers for your realm:

kdc.insat.tn

<Ok>
```

1.4- Setting up Kerberos realm:

This can be done using : `krb5_newrealm`

1.5-then we need to grant the admin privileges by editing /etc/krb5kdc/kadm5.acl:

we just need to uncomment */admin* :

```
  GNU nano 6.2                    kadm5.acl *
# This file Is the access control list for krb5 administration.
# When this file is edited run service krb5-admin-server restart to activate
# One common way to set up Kerberos administration is to allow any principal
# ending in /admin  is given full administrative rights.
# To enable this, uncomment the following line:
 */admin@insat.tn *
```

then restart krb5 server:

```
systemctl restart krb5-admin-server
```

2- Creating Principals:

2.1 In order to create Kerberos principals we should first login as admin then:

```
kadmin.local                                # login as local admin
addprinc root/admin                         # add admin principal
addprinc -randkey host/kdc.insat.tn      # add host principal
ktadd host/kdc.insat.tn              # generate host principal keytab line
```

```
kadmin.local:  addprinc root/admin
No policy specified for root/admin@INSAT.TN; defaulting to no policy
Enter password for principal "root/admin@INSAT.TN":
Re-enter password for principal "root/admin@INSAT.TN":
Principal "root/admin@INSAT.TN" created.
kadmin.local:
```

```
kadmin.local:  addprinc -randkey host/kdc.insat.tn
No policy specified for host/kdc.insat.tn@INSAT.TN; defaulting to no policy
Principal "host/kdc.insat.tn@INSAT.TN" created.
kadmin.local:
```

```
kadmin.local:  listprincs
K/M@INSAT.TN
host/kdc.insat.tn@INSAT.TN
kadmin/admin@INSAT.TN
kadmin/changepw@INSAT.TN
krbtgt/INSAT.TN@INSAT.TN
root/admin@INSAT.TN
user@INSAT.TN
kadmin.local:
```

```
kadmin.local:  ktadd host/kdc.insat.tn
Entry for principal host/kdc.insat.tn with kvno 2, encryption type aes256-cts-hm
ac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kdc.insat.tn with kvno 2, encryption type aes128-cts-hm
ac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
kadmin.local:
```

2.2- To add entry to current key list then write current keylist to the keytab:

```
ktutil
```

```
addent -password -p root/admin@INSAT.TN -k 1 -e aes256-cts-hmac-sha1-96
wkt /etc/krb5kdc/kadm5.acl
```

```
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
```

```
Keytab name: FILE:kadm5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   2 host/kdc.insat.tn@INSAT.TN
   2 host/kdc.insat.tn@INSAT.TN
   1 root/admin@INSAT.TN
```

and the same for the host :

```
ktutil
```

```
addent -password -p host/kdc.insat.tn -k 1 -e aes256-cts-hmac-sha1-96
wkt kadm5.acl
```

```
ktutil:  addent -password -p host/kdc.insat.tn -k 1 -e aes256-cts-hmac-sha1-96
Password for host/kdc.insat.tn@INSAT.TN:
ktutil:  wkt kadm5.keytab
ktutil:  q
```

```
root@kdc:/etc/krb5kdc# file kadm5.keytab
kadm5.keytab: Kerberos Keytab file, realm=INSAT.TN, principal=ho
 type=1, date=Wed Jan 17 11:14:04 2024, kvno=2
root@kdc:/etc/krb5kdc# klist -k kadm5.keytab
Keytab name: FILE:kadm5.keytab
KVNO Principal
---- --------------------------------------------------------------
   2 host/kdc.insat.tn@INSAT.TN
   2 host/kdc.insat.tn@INSAT.TN
   1 root/admin@INSAT.TN
   1 host/kdc.insat.tn@INSAT.TN
```

2.3- add a user principal:

```
kadmin.local
addprinc user
```

```
kadmin.local:  get_principal user
Principal: user@INSAT.TN
Expiration date: [never]
Last password change: Wed Jan 17 11:36:44 WAT 2024
Password expiration date: [never]
Maximum ticket life: 0 days 10:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Jan 17 11:36:44 WAT 2024 (root/admin@INSAT.TN)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 2
Key: vno 1, aes256-cts-hmac-sha1-96
Key: vno 1, aes128-cts-hmac-sha1-96
MKey: vno 1
Attributes: REQUIRES_PRE_AUTH
Policy: [none]
```

To force users to change their password upon the first login, you can use the -pwexpire option:

```
modprinc -pwexpire now user
addpol -maxlife time
```

```
┌──(user㉿client)-[~]
└─$ kinit
Password for user@INSAT.TN:
Password expired.  You must change it now.
Enter new password:
```

Section 2: Authentication with SSH:

1- first we need to install krb5-user

```
apt install krb5-user
```

```
  GNU nano 7.2                    /etc/krb5.conf
[libdefaults]
        default_realm = INSAT.TN

# The following krb5.conf variables are only for MIT Kerberos.
        kdc_timesync = 1
        ccache_type = 4
        forwardable = true
        proxiable = true
        rdns = false


# The following libdefaults parameters are only for Heimdal Kerberos.
        fcc-mit-ticketflags = true

[realms]
        INSAT.TN = {
                kdc = kdc.insat.tn
                admin_server = kdc.insat.tn
        }
```

2-install openssh

3- Configure SSH for GSSAPI Authentication:

```
  GNU nano 7.2                    /etc/ssh/ssh_config

Host *
#   ForwardAgent no
#   ForwardX11 no
#   ForwardX11Trusted yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
   GSSAPIAuthentication yes
   GSSAPIDelegateCredentials yes
#   GSSAPIKeyExchange no
#   GSSAPITrustDNS no
#   BatchMode no
```

```
  GNU nano 7.2                    /etc/ssh/sshd_config
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
```

4-restart ssh service

```
systemctl restart sshd /ssh
```

5-the same thing for the kdc machine

6-create a new user in the client machine with the same name of the principale created in kdc :

```
adduser user
su -l user
```

```
┌──(root㉿client)-[/home/kali]
└─# adduser user
info: Adding user `user' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `user' (1001) ...
info: Adding new user `user' (1001) with group `user (1001)' ...
info: Creating home directory `/home/user' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `user' to supplemental / extra groups `users' ...
info: Adding user `user' to group `users' ...

┌──(root㉿client)-[/home/kali]
└─# su -l user
┌──(user㉿client)-[~]
└─$
```

7- run `kinit` in order to send request to get TGT

```
┌──(user㉿client)-[~]
└─$ klist
klist: No credentials cache found (filename: /tmp/krb5cc_1001)

┌──(user㉿client)-[~]
└─$
```

```
┌──(user㉿client)-[~]
└─$ kinit
Password for user@INSAT.TN:

┌──(user㉿client)-[~]
└─$ klist
Ticket cache: FILE:/tmp/krb5cc_1001
Default principal: user@INSAT.TN

Valid starting       Expires              Service principal
01/17/2024 08:02:48  01/17/2024 18:02:48  krbtgt/INSAT.TN@INSAT.TN
        renew until 01/18/2024 08:02:41
```

8-then run `ssh kdc.insat.tn`

The actual SSH connection (ssh kdc.insat.tn) involves the TGS-REQ and TGS-REP steps, where the client requests a service ticket for the SSH service.

then the ssh connection should be setted successfully without a password.

```
┌──(user㉿client)-[~]
└─$ kinit
Password for user@INSAT.TN:

┌──(user㉿client)-[~]
└─$ klist
Ticket cache: FILE:/tmp/krb5cc_1001
Default principal: user@INSAT.TN

Valid starting       Expires              Service principal
01/17/2024 16:21:53  01/18/2024 02:21:53  krbtgt/INSAT.TN@INSAT.TN
        renew until 01/18/2024 16:21:46

┌──(user㉿client)-[~]
└─$ ssh kdc.insat.tn
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

180 updates can be applied immediately.
122 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Jan 17 22:21:40 2024 from 192.168.56.102
user@kdc:~$
```

```
root@kdc:/home/firas# w
 22:24:31 up 13:22,  8 users,  load average: 0.01, 0.10, 0.06
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
firas    tty2     tty2             09:05   13:22m  0.28s  0.15s /usr/libexec/gn
firas    pts/1    -                11:20   10:02m  0.60s  0.55s sudo su
firas    pts/3    -                11:24    0.00s  1.64s  1.41s sudo su
firas    pts/5    -                12:22   12.00s  0.45s  0.46s sudo su
firas    pts/7    -                14:11    1:21m  0.13s  0.90s sudo su
firas    pts/8    192.168.56.101   18:22    4:00m  0.69s  0.13s sshd: firas [pr
firas    pts/9    192.168.56.101   18:24    0.00s  0.38s  0.63s sudo su
user     pts/11   192.168.56.102   22:22    2:23   0.03s  0.03s -bash
root@kdc:/home/firas#
```