Cybersecurity Project

Team members :

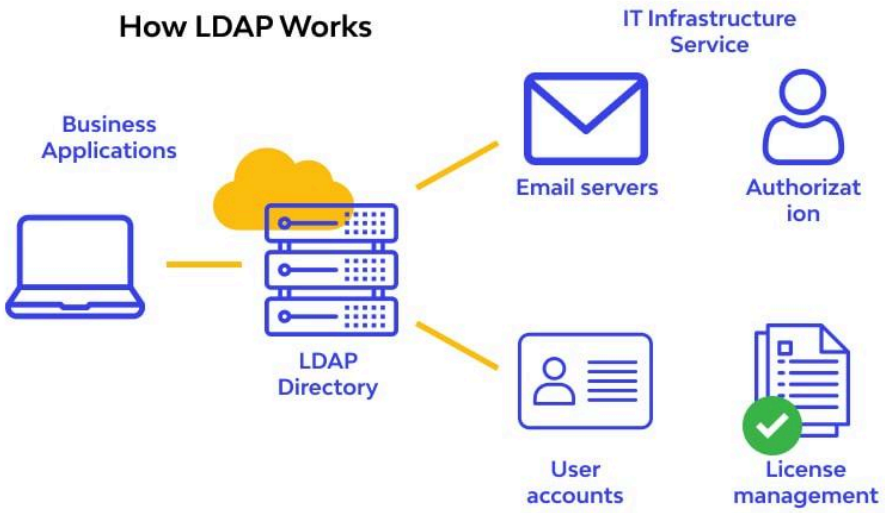
Tasnim Dakhli - Achref Saidi - Ali Doggaz - Mahdi Ghorbel

Table of Content

Authentication with OpenLDAP, SSH, Apache, OpenVPN

1. OpenLDAP Configuration

OpenLDAP is an open-source implementation of the Lightweight Directory Access Protocol (LDAP). It serves as a centralized directory service, organizing and storing information in a hierarchical structure.
Commonly used for authentication, authorization, and managing user-related data
Example : Someone within the HR department wants to do two things: Send an email to a recent hire and print a copy of that conversation on a new printer. If LDAP is set up properly that employee doesn't need to talk with IT to complete the tasks.



Now let's see how we can make that possible :

On the server side :

Step 1 : check full hostname :

```
tasnim@tasnim:~$ hostname -f
tasnim.gl4.local
```

Don’t turn a blind eye on it. Trust me I have been there. Here is a quick way to configure it : just modify the **/etc/hosts** file.

```
tasnim@tasnim:/etc/bind$ sudo cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       tasnim.security.local tasnim
192.168.56.105  tasnim.security.local tasnim
```

Step2 :

```
tasnim@tasnim:~$ sudo apt install apache2 php php-cgi libapache2-mod-php php-mbstring php-common php-pear -y
[sudo] password for tasnim:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

```
tasnim@tasnim:~$ sudo apt install slapd ldap-utils -y
```

Verify : **sudo slapcat**

```
tasnim@tasnim:~$ sudo slapcat
dn: dc=gl4,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: gl4.local
dc: gl4
structuralObjectClass: organization
entryUUID: c57bcf84-4744-103e-89ba-5f5e4ba3bf65
creatorsName: cn=admin,dc=gl4,dc=local
createTimestamp: 20240114162152Z
entryCSN: 20240114162152.763415Z#000000#000#000000
modifiersName: cn=admin,dc=gl4,dc=local
modifyTimestamp: 20240114162152Z
```

Step 3 : Install LAM (Ldap Account Manager)

```
tasnim@tasnim:~$ sudo apt -y install ldap-account-manager
```

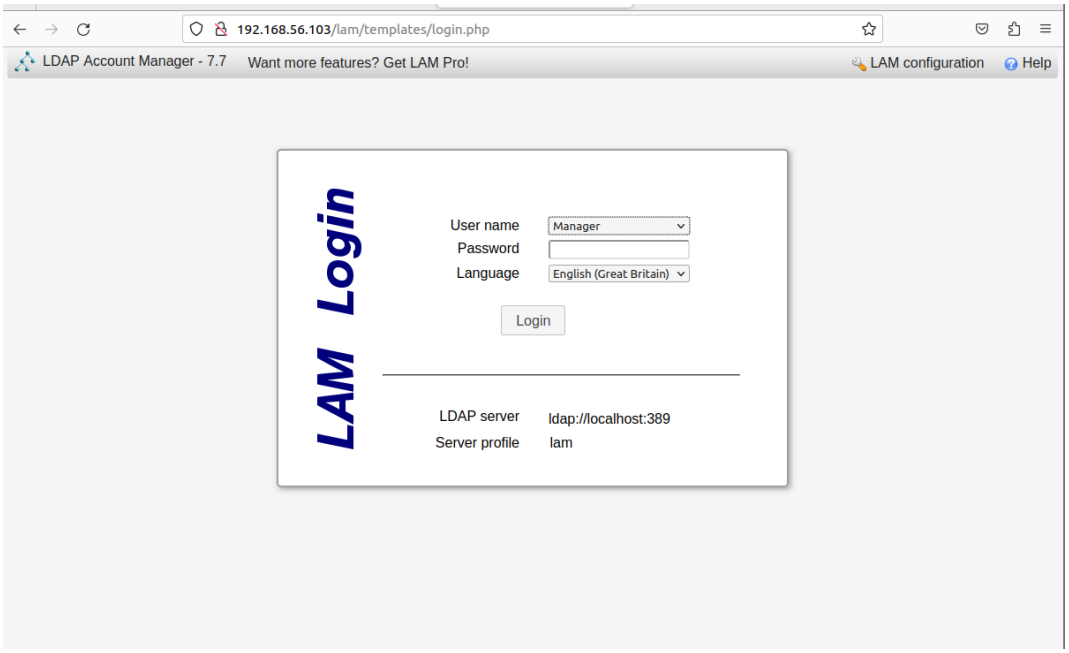
Step 4 : Enable the configuration file(s) related to PHP FastCGI

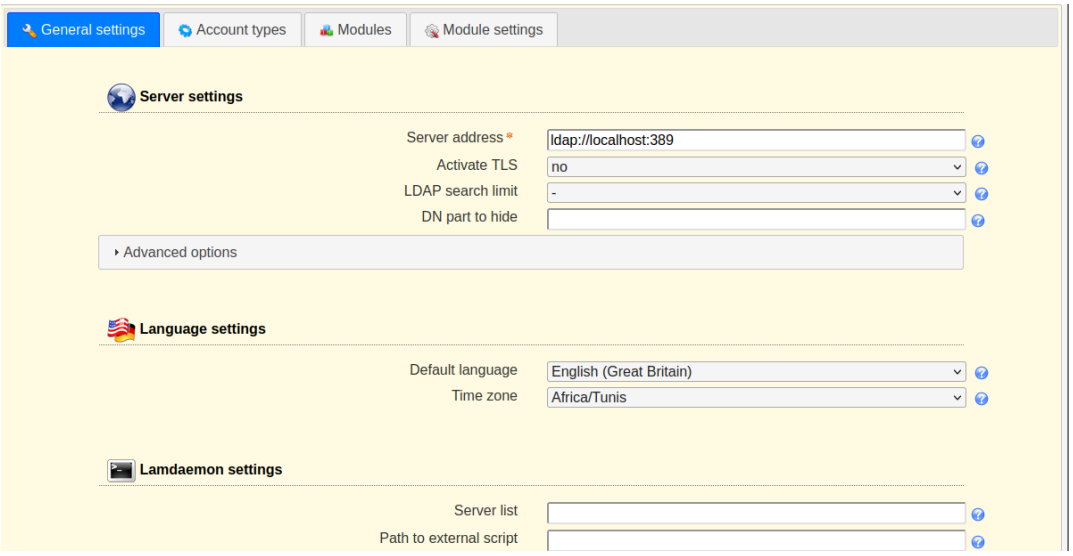
```
tasnim@tasnim:~$ sudo a2enconf php*-cgi
Enabling conf php8.1-cgi.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

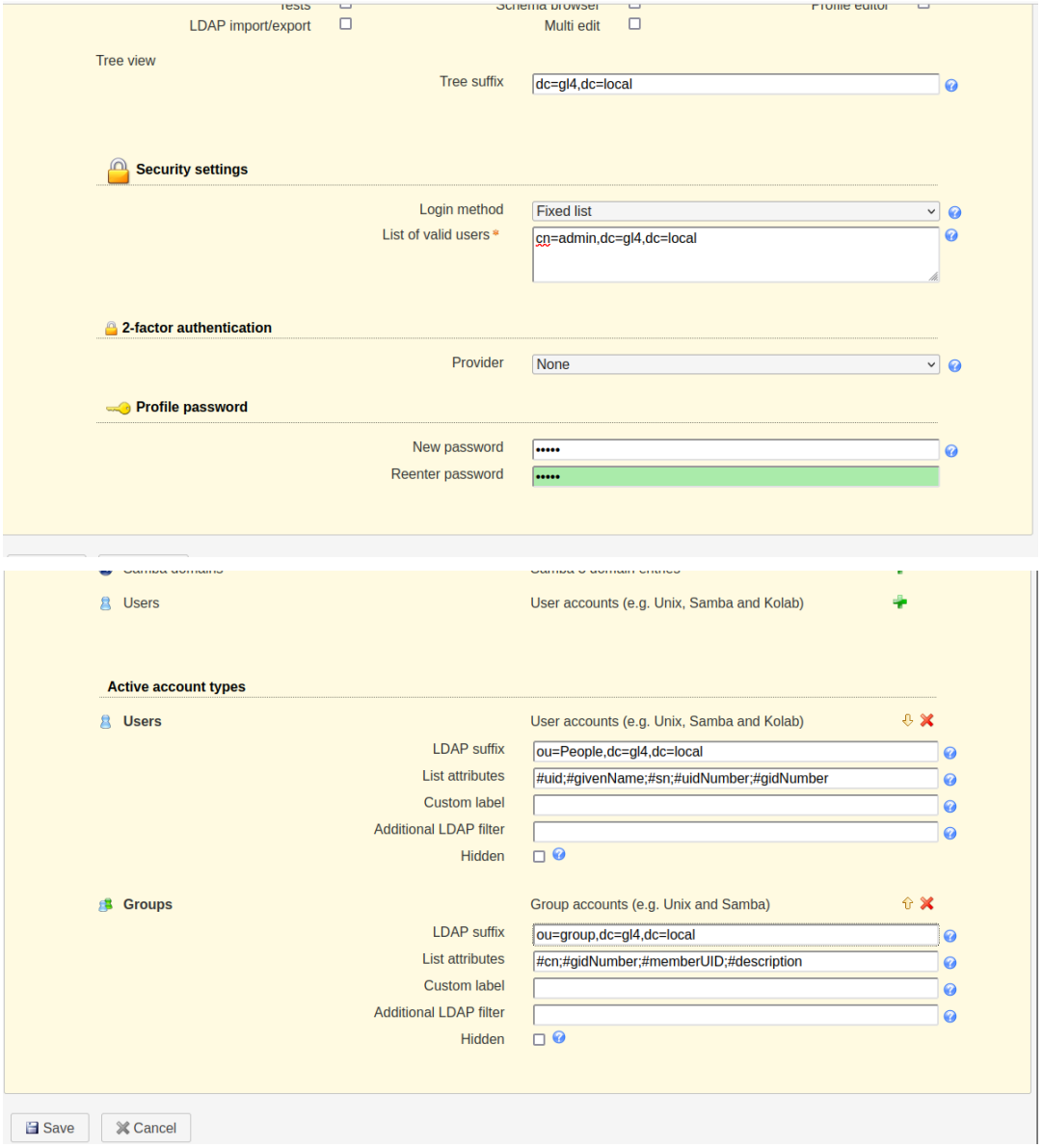
Step 5 : Restart the Apache server

```
tasnim@tasnim:~$ sudo systemctl restart apache2
tasnim@tasnim:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
tasnim@tasnim:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-01-14 17:26:36 CET; 47s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 18458 (apache2)
      Tasks: 6 (limit: 4599)
     Memory: 13.8M
        CPU: 61ms
     CGroup: /system.slice/apache2.service
             ├─18458 /usr/sbin/apache2 -k start
             ├─18459 /usr/sbin/apache2 -k start
             ├─18460 /usr/sbin/apache2 -k start
             ├─18461 /usr/sbin/apache2 -k start
             ├─18462 /usr/sbin/apache2 -k start
             └─18463 /usr/sbin/apache2 -k start

17:26:36 14 جانفي tasnim systemd[1]: Starting The Apache HTTP Server...
17:26:36 14 جانفي tasnim systemd[1]: Started The Apache HTTP Server.
```

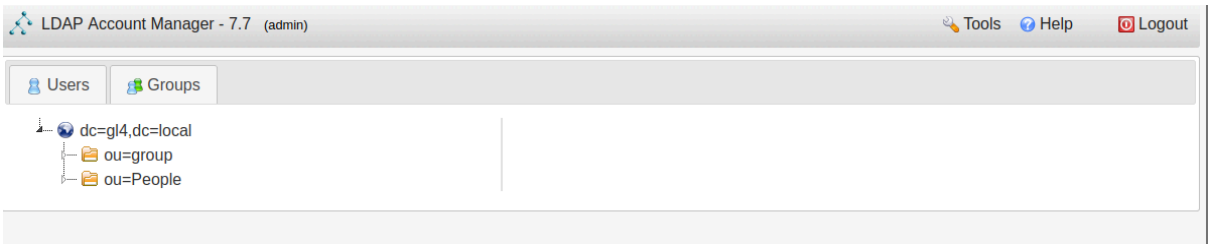
Step 6 : open LAM : **youripaddress/lam**



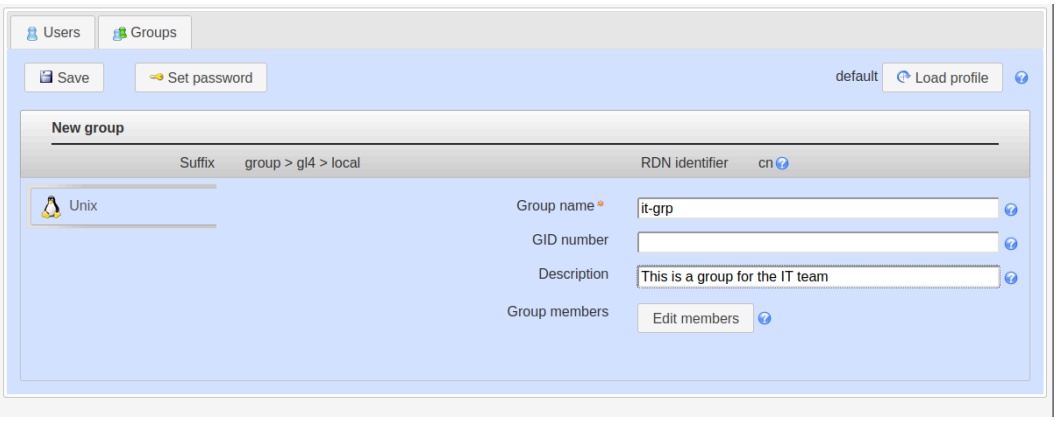


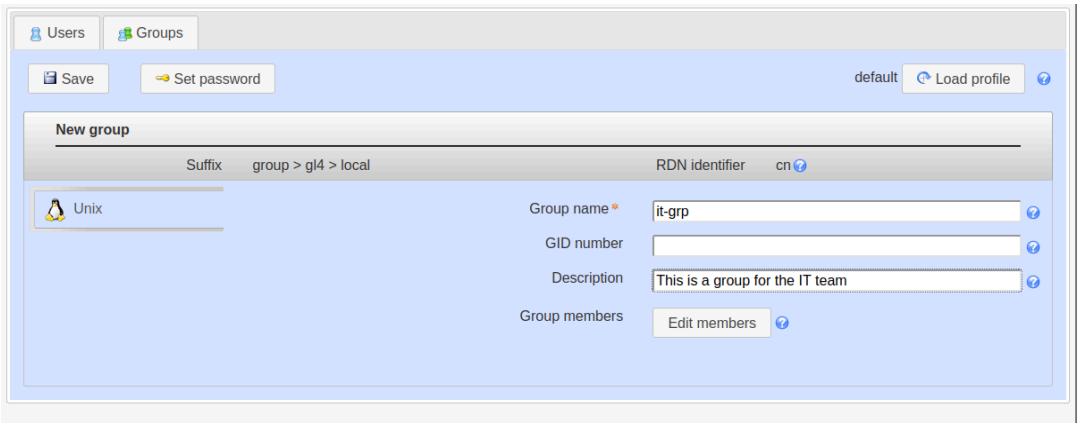


This is how your tree would look like so far : (you can find it under Tools)

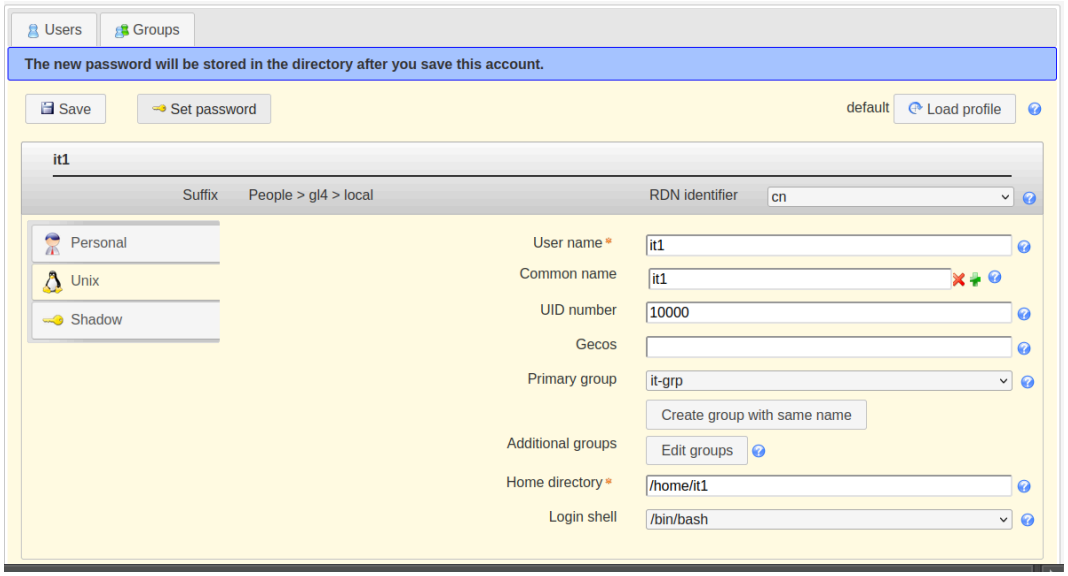


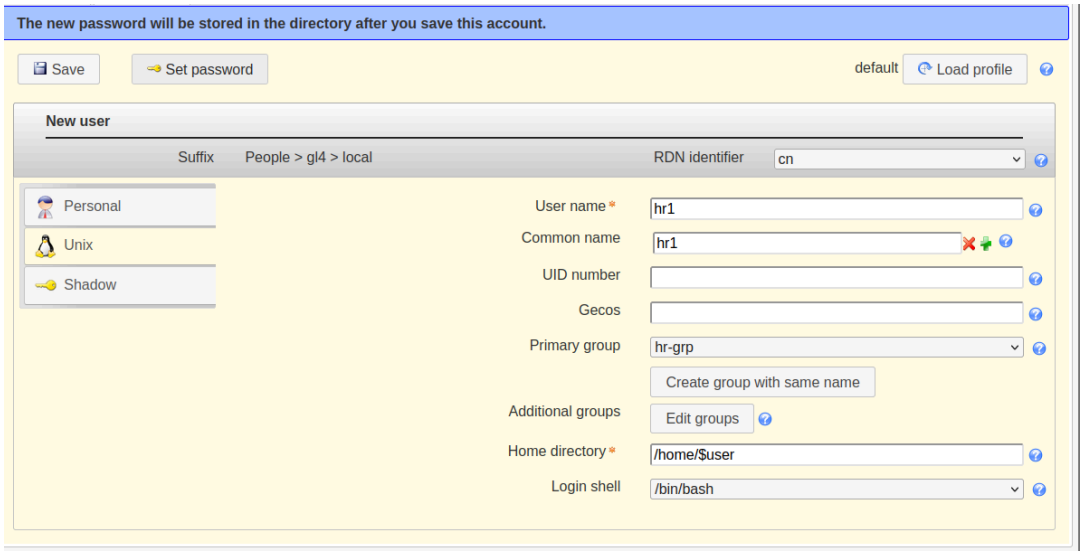
Group creation : you can create as many groups as you want, but for the sake of this Tutorial we will create 2 : **it-grp** and **hr-grp**



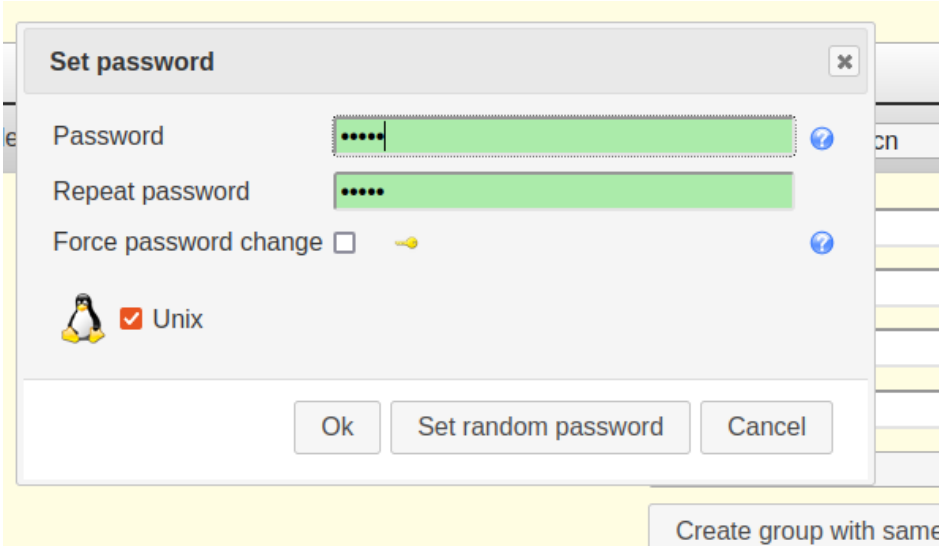


User Creation :

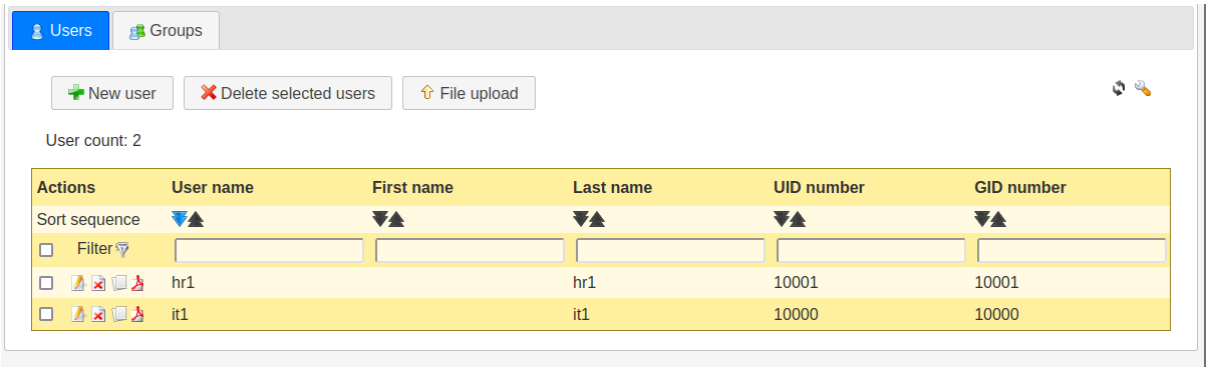


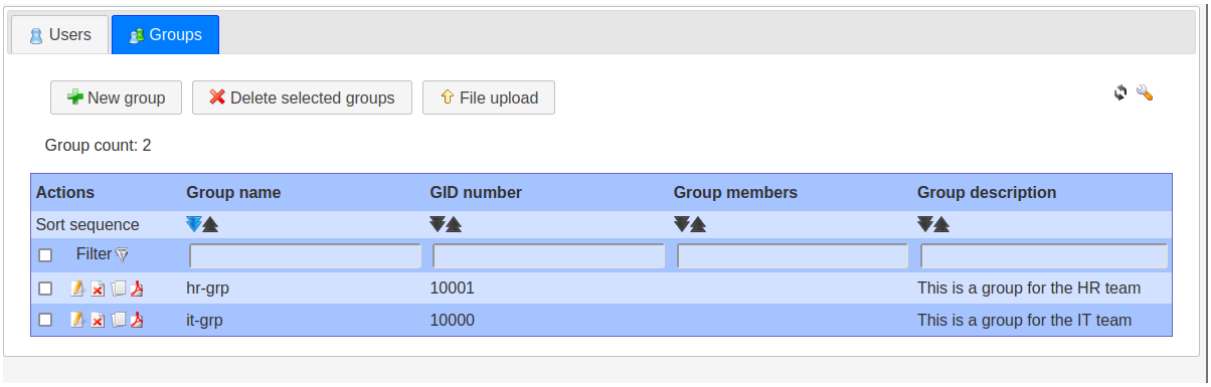


Setting up passwords :



After Creation :





On the Client side :

Now that we finished the configuration on the Server side, we will need to do it on the Client side.

Prerequisites :

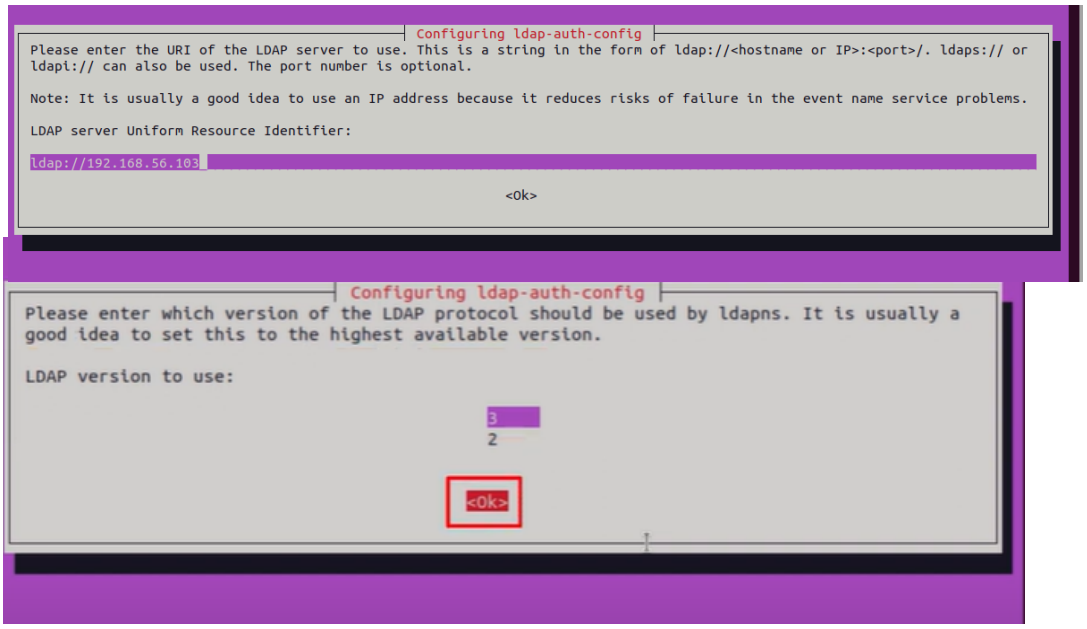
1. Make sure OpenLDAP service is running on the Server side :
   sudo systemctl status slapd
2. Add the IP and FQDN of LDAP server to file **/etc/hosts** of the client
3. Make sure you can ping the LDAP server

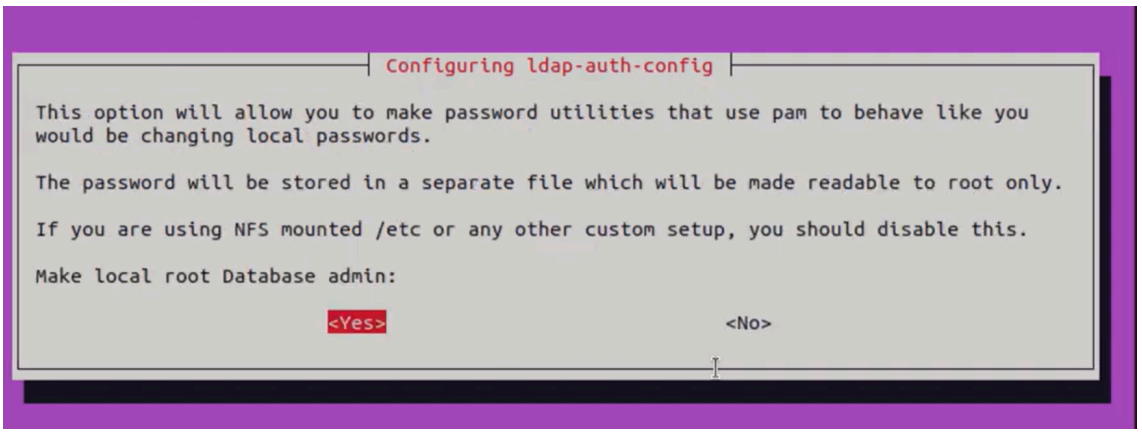
Now let's dive into the installation:

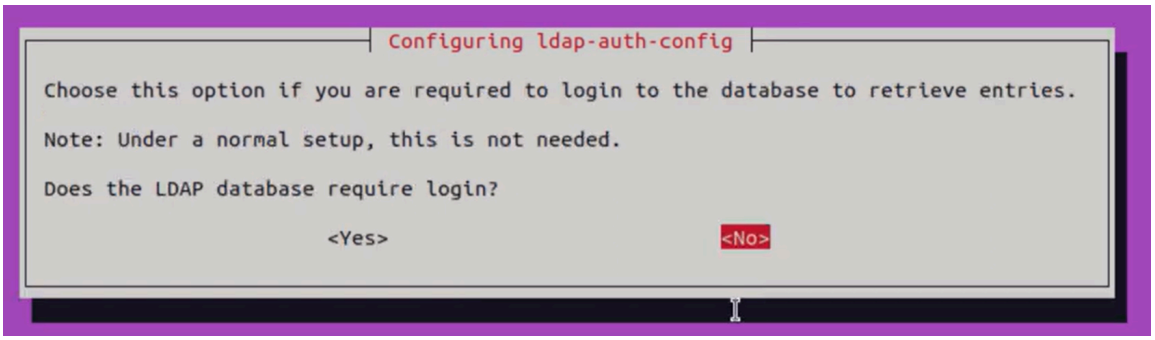
Step 1 : update the system : **sudo apt update**

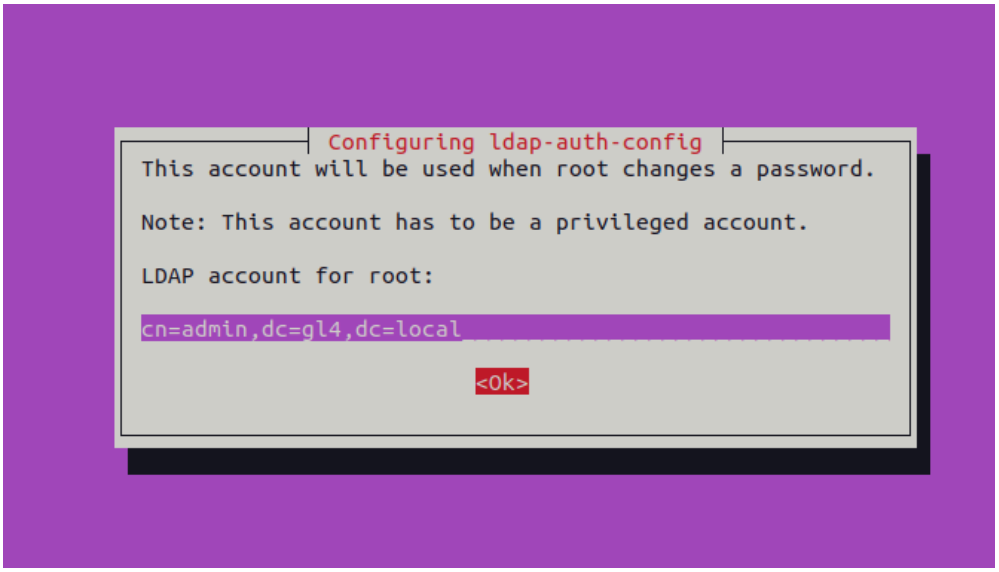
Step 2 : **sudo apt install libnss-ldap libpam-ldap ldap-utils nscd -y**

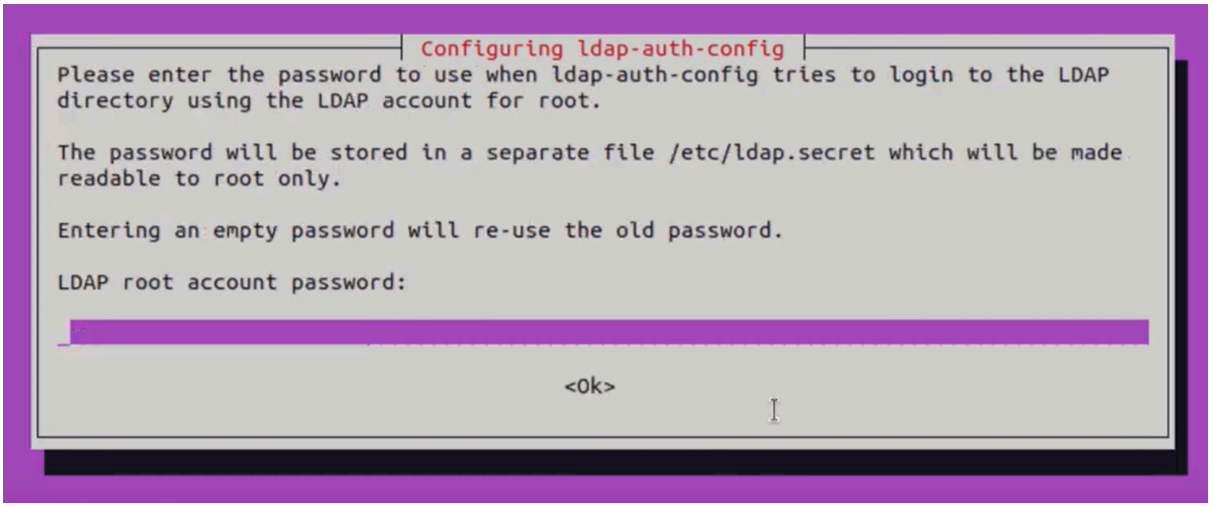
Step 3 :




Configuring ldap-auth-config
This option will allow you to make password utilities that use pam to behave like you would be changing local passwords.
The password will be stored in a separate file which will be made readable to root only.
If you are using NFS mounted /etc or any other custom setup, you should disable this.
Make local root Database admin:
<Yes>
<No>


Configuring ldap-auth-config
Choose this option if you are required to login to the database to retrieve entries.
Note: Under a normal setup, this is not needed.
Does the LDAP database require login?
<Yes>
<No>


Configuring ldap-auth-config
This account will be used when root changes a password.
Note: This account has to be a privileged account.
LDAP account for root:
cn=admin,dc=gl4,dc=local
<Ok>



Step 4: change file **/etc/nsswitch.conf**

```
# /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:         compat systemd ldap
group:          compat systemd ldap
shadow:         compat
gshadow:        files

hosts:          files mdns4_minimal [NOTFOUND=return] dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Step 5: change file **/etc/pam.d/common-password**

```
#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords.  The default is pam_unix.

# Explanation of pam_unix options:
# The "yescrypt" option enables
#hashed passwords using the yescrypt algorithm, introduced in Debian
#11.  Without this option, the default is Unix crypt.  Prior releases
#used the option "sha512"; if a shadow password hash will be shared
#between Debian 11 and older releases replace "yescrypt" with "sha512"
#for compatibility .  The "obscure" option replaces the old
#`OBSCURE_CHECKS_ENAB' option in login.defs.  See the pam_unix manpage
#for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so retry=3
password        [success=3 default=ignore]      pam_unix.so obscure use_authtok try_first_pass yescrypt
password        sufficient                      pam_sss.so use_authtok
password        [success=1 user_unknown=ignore default=die]     pam_ldap.so try_first_pass
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so

^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
```

Step 5: change file **/etc/pam.d/common-session**

```
 GNU nano 6.2                                  /etc/pam.d/common-session *
#
# /etc/pam.d/common-session - session-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define tasks to be performed
# at the start and end of interactive sessions.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
session [default=1]                     pam_permit.so
# here's the fallback if no module succeeds
session requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required                        pam_permit.so
# The pam_umask module will set the umask according to the system default in
# /etc/login.defs and user settings, solving the problem of different
# umask settings with different shells, display managers, remote sessions etc.
# See "man pam_umask".
session optional                        pam_umask.so
# and here are more per-package modules (the "Additional" block)
session required        pam_unix.so
session optional                        pam_sss.so
session optional                        pam_ldap.so
session optional        pam_systemd.so
# end of pam-auth-update config
session optional pam_mkhomedir.so skel=/etc/skel umask=077
```

Step 6 : restart nscd

sudo systemctl restart nscd **sudo systemctl enable nscd**

Verifying users' authentication

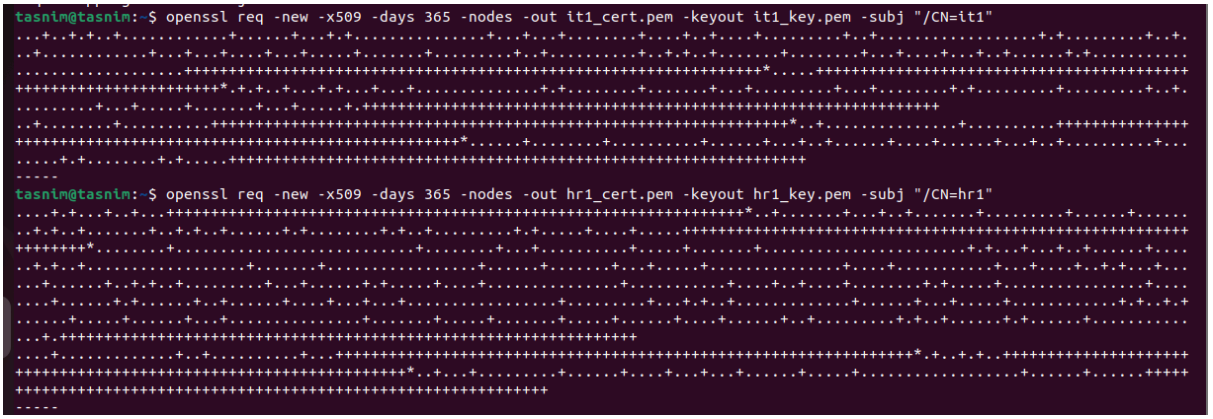
```
client@client:~$ ldapsearch -x -H ldap://192.168.56.103 -b "dc=gl4,dc=local"
# extended LDIF
#
# LDAPv3
# base <dc=gl4,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# gl4.local
dn: dc=gl4,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: gl4.local
dc: gl4

# People, gl4.local
dn: ou=People,dc=gl4,dc=local
objectClass: organizationalUnit
ou: People

# group, gl4.local
dn: ou=group,dc=gl4,dc=local
objectClass: organizationalUnit
ou: group

# it-grp, group, gl4.local
dn: cn=it-grp,ou=group,dc=gl4,dc=local
objectClass: posixGroup
description: This is a group for the IT team
gidNumber: 10000
cn: it-grp

# hr-grp, group, gl4.local
dn: cn=hr-grp,ou=group,dc=gl4,dc=local
objectClass: posixGroup
description: This is a group for the HR team
gidNumber: 10001
cn: hr-grp
```

```
client@client:~$ sudo login
client login: hr1
Password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-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.
134 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


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.

Creating directory '/home/hr1'.
hr1@client:~$
```

Step 6 : celebrate!

Adding x509 certificates to users :

Step 1 : generate self signed certificates :



Step 2 : create a file : cert.ldif

```
  GNU nano 6.2                          cert.ldif *
dn: uid=it1,ou=people,dc=gl4,dc=local
changetype: modify
add: userCertificate;binary
userCertificate;binary:: MIIC/TCCAeWgAwIBAgIUWO1DsC6Sag9THMQA+Qy4E0mvgOswDQYJKo>

n: uid=hr1,ou=people,dc=gl4,dc=local
changetype: modify
add: userCertificate;binary
userCertificate;binary:: MIIC/TCCAeWgAwIBAgIUX75C2MpkMngMVr9THA95Hopjmq8wDQYJKo>
```

Step 3 :

```
tasnim@tasnim:~$ ldapmodify -x -D "cn=admin,dc=gl4,dc=local" -W -f cert.ldif
Enter LDAP Password:
modifying entry "cn=it1,ou=People,dc=gl4,dc=local"
```

Testing the secure part of LDAP with LDAPS :

LDAPS is a protocol that uses SSL (Secure Sockets Layer) or TLS (Transport Layer Security) to encrypt and authenticate the data exchanged between an LDAP client and an LDAP server. It enhances the security and privacy of your LDAP data

Advantages of using LDAPS :

- **Confidentiality through data encryption :** This prevents anyone from reading or altering it in transit, which can protect you from identity theft, data breaches, or compliance violations
- **Authentication Security :** This ensures that you are communicating with the intended party, and not with an impostor or a rogue server. This can protect you from phishing, spoofing, or man-in-the-middle attacks.
- **Data Integrity:** The SSL/TLS protocols include mechanisms to detect and reject any data tampering or corruption that may occur during transit.
- **Meeting Security Standards:** LDAPS can help you meet the security standards and regulations that apply to your industry or organization, such as HIPAA, PCI-DSS, or GDPR.

Step 1 : Generating self signed certificate

Create new folder ssl-ldap

- **mkdir ssl-ldap**
- **cd ssl-ldap**

Generate certificate :

```
tasnim@tasnim:~/ssl-ldap$ openssl genrsa -aes128 -out tasnim.gl4.local.key 4096
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

```
tasnim@tasnim:~/ssl-ldap$ ll
total 12
drwxrwxr-x  2 tasnim tasnim 4096 18:35 14  جانفي ./
drwxr-x--- 15 tasnim tasnim 4096 18:33 14  جانفي ../
-rw-------  1 tasnim tasnim 3434 18:35 14  جانفي tasnim.gl4.local.key
tasnim@tasnim:~/ssl-ldap$ openssl rsa -in tasnim.gl4.local.key -out tasnim.gl4.local.key
Enter pass phrase for tasnim.gl4.local.key:
writing RSA key
tasnim@tasnim:~/ssl-ldap$
```

```
writing RSA key
tasnim@tasnim:~/ssl-ldap$ openssl req -new -days 3650 -key tasnim.gl4.local.key -out tasnim.gl4.local.csr
Ignoring -days without -x509; not generating a certificate
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) []:Tunis
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Insat
Organizational Unit Name (eg, section) []:gl4
Common Name (e.g. server FQDN or YOUR name) []:tasnim.gl4.local
Email Address []:tasnimdakhlitd@gmail.com

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

```
tasnim@tasnim:~/ssl-ldap$ sudo openssl x509 -in tasnim.gl4.local.csr -out tasnim.gl4.local.crt -req -signkey tasnim.gl4.local.key -d
ays 3650
Certificate request self-signature ok
subject=C = TN, ST = Tunis, L = Tunis, O = Insat, OU = gl4, CN = tasnim.gl4.local, emailAddress = tasnimdakhlitd@gmail.com
```

Step 2 : move them to /etc/ldap/sasl2/

```
tasnim@tasnim:~/ssl-ldap$ sudo cp tasnim.gl4.local.key /etc/ldap/sasl2
tasnim@tasnim:~/ssl-ldap$ sudo cp tasnim.gl4.local.crt /etc/ldap/sasl2
tasnim@tasnim:~/ssl-ldap$ sudo cp /etc/ssl/certs/ca-certificates.crt /etc/ldap/sasl2/
tasnim@tasnim:~/ssl-ldap$ ll /etc/ldap/sasl2/
total 220
drwxr-xr-x 2 root root   4096 18:46 14  جانفي ./
drwxr-xr-x 5 root root   4096 17:21 14  جانفي ../
-rw-r--r-- 1 root root 208567 18:46 14  جانفي ca-certificates.crt
-rw-r--r-- 1 root root   2021 18:45 14  جانفي tasnim.gl4.local.crt
-rw------- 1 root root   3272 18:45 14  جانفي tasnim.gl4.local.key
```

Step 3 : change ownership to openldap

```
tasnim@tasnim:~/ssl-ldap$ sudo chown -R openldap:openldap /etc/ldap/sasl2
tasnim@tasnim:~/ssl-ldap$ ll /etc/ldap/sasl2/
total 220
drwxr-xr-x 2 openldap openldap   4096 18:46 14  جانفي ./
drwxr-xr-x 5 root     root       4096 17:21 14  جانفي ../
-rw-r--r-- 1 openldap openldap 208567 18:46 14  جانفي ca-certificates.crt
-rw-r--r-- 1 openldap openldap   2021 18:45 14  جانفي tasnim.gl4.local.crt
-rw------- 1 openldap openldap   3272 18:45 14  جانفي tasnim.gl4.local.key
tasnim@tasnim:~/ssl-ldap$
```

Step 4: create LDAP configuration file called ssl-ldap.ldif , configure LDAP server to use SSL certificates

```
  GNU nano 6.2                                          SSL-LDAP.ldif
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/sasl2/ca-certificates.crt
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/sasl2/tasnim.gl4.local.crt
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/sasl2/tasnim.gl4.local.key
```

```
tasnim@tasnim:~/ssl-ldap$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f SSL-LDAP.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
```

Step 5 : Edit /etc/default/slapd.conf (add ldaps:///)

```
  GNU nano 6.2                                          /etc/default/slapd
# Default location of the slapd.conf file or slapd.d cn=config directory. If
# empty, use the compiled-in default (/etc/ldap/slapd.d with a fallback to
# /etc/ldap/slapd.conf).
SLAPD_CONF=

# System account to run the slapd server under. If empty the server
# will run as root.
SLAPD_USER="openldap"

# System group to run the slapd server under. If empty the server will
# run in the primary group of its user.
SLAPD_GROUP="openldap"

# Path to the pid file of the slapd server. If not set the init.d script
# will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.d by
# default)
SLAPD_PIDFILE=

# slapd normally serves ldap only on all TCP-ports 389. slapd can also
# service requests on TCP-port 636 (ldaps) and requests via unix
# sockets.
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldap:/// ldapi:/// ldaps:///"
```

Step 6 : Edit /etc/ldap/ldap.conf

```
  GNU nano 6.2                                          /etc/ldap/ldap.conf *
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASE   dc=example,dc=com
#URI    ldap://ldap.example.com ldap://ldap-provider.example.com:666

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never

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

Step 7: restart slapd

```
tasnim@tasnim:~/ssl-ldap$ sudo systemctl restart slapd
```

Step 8: verify

```
tasnim@tasnim:~/ssl-ldap$ ldapsearch -x -H ldaps://192.168.56.103 -b "dc=gl4,dc=local"
# extended LDIF
#
# LDAPv3
# base <dc=gl4,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# gl4.local
dn: dc=gl4,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: gl4.local
dc: gl4

# People, gl4.local
dn: ou=People,dc=gl4,dc=local
objectClass: organizationalUnit
ou: People

# group, gl4.local
dn: ou=group,dc=gl4,dc=local
objectClass: organizationalUnit
ou: group

# it-grp, group, gl4.local
dn: cn=it-grp,ou=group,dc=gl4,dc=local
objectClass: posixGroup
description: This is a group for the IT team
gidNumber: 10000
cn: it-grp

# hr-grp, group, gl4.local
dn: cn=hr-grp,ou=group,dc=gl4,dc=local
objectClass: posixGroup
description: This is a group for the HR team
gidNumber: 10001
```

```
tasnim@tasnim:~/ssl-ldap$ netstat -antup | grep -i 636
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 0.0.0.0:636             0.0.0.0:*               LISTEN      -
tcp6       0      0 :::636                  :::*                    LISTEN      -
tasnim@tasnim:~/ssl-ldap$ nano SSL-LDAP.ldif
```

LDAPS listening on port 636

2. SSH Authentication

SSH The Secure Shell protocol is a network protocol used for secure communication over an unsecured network. SSH uses cryptography to authenticate and encrypt connections between devices.SSH is often used for controlling servers remotely, for managing infrastructure, and for transferring files protecting them from eavesdropping and tampering.

1. Authentication SSH via OpenLDAP:

   sudo nano /etc/nsswitch.conf

```
 GNU nano 6.2                                  /etc/nsswitch.conf *
# /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:         compat systemd ldap
group:          compat systemd ldap
shadow:         compat ldap
gshadow:        files systemd

hosts:          files mdns4_minimal [NOTFOUND=return] dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

sudo nano /etc/pam.d/common-auth

```
 GNU nano 6.2                                    common-auth
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
auth    [success=3 default=ignore]      pam_unix.so nullok
auth    [success=2 default=ignore]      pam_sss.so use_first_pass
auth    [success=1 default=ignore]      pam_ldap.so use_first_pass
# here's the fallback if no module succeeds
auth    requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
auth    optional                        pam_cap.so
# end of pam-auth-update config
```

sudo nano /etc/pam.d/common-account

```
client@client:/etc/pam.d$ sudo cat common-account
#
# /etc/pam.d/common-account - authorization settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired in /etc/shadow.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.
#

# here are the per-package modules (the "Primary" block)
account [success=2 new_authtok_reqd=done default=ignore]        pam_unix.so
account [success=1 default=ignore]      pam_ldap.so
# here's the fallback if no module succeeds
account requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
account required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
account sufficient                      pam_localuser.so
account [default=bad success=ok user_unknown=ignore]    pam_sss.so
# end of pam-auth-update config
```

sudo cat /etc/pam.d/common-password

```
client@client:/etc/pam.d$ sudo cat common-password
#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords.  The default is pam_unix.

# Explanation of pam_unix options:
# The "yescrypt" option enables
#hashed passwords using the yescrypt algorithm, introduced in Debian
#11.  Without this option, the default is Unix crypt.  Prior releases
#used the option "sha512"; if a shadow password hash will be shared
#between Debian 11 and older releases replace "yescrypt" with "sha512"
#for compatibility .  The "obscure" option replaces the old
#`OBSCURE_CHECKS_ENAB' option in login.defs.  See the pam_unix manpage
#for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so retry=3
password        [success=3 default=ignore]      pam_unix.so obscure use_authtok try_first_pass yescrypt
password        sufficient                      pam_sss.so use_authtok
password        [success=1 user_unknown=ignore default=die]     pam_ldap.so try_first_pass
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so
# end of pam-auth-update config
```

2. Restrict SSH access to users in the appropriate group in OpenLDAP:

   Add the following line to **/etc/ssh/sshd_config** “**AllowGroups it-grp**” to only allow users from it group to access SSH.

   sudo systemctl restart ssh

3. Test for an authorized user and an unauthorized user:

   Test the access to SSH on a distant machine using **“it1”** a user that belongs to **it-grp**

   ssh it1@192.168.56.104

```
[-w local_tun[:remote_tun]] destination [command [argument ...]]
tasnim@tasnim:~$ ssh it1@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:L3UpNAPzWs7AZJMft8Usvk/n9JztzktHbaI8jkWlD2I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.104' (ED25519) to the list of known hosts.
it1@192.168.56.104's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-14-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.

56 updates can be applied immediately.
9 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


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.

Last login: Mon Jan 15 20:49:10 2024
```

This user is allowed to login.

Test again with another user **“hr1”** a user that doesn’t belong to **it-grp**.

```
it1@client:~$ ssh hr1@192.168.56.104
The authenticity of host '192.168.56.104 (192.168.56.104)' can't be established.
ED25519 key fingerprint is SHA256:L3UpNAPzWs7AZJMft8Usvk/n9JztzktHbaI8jkWlD2I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.104' (ED25519) to the list of known hosts.
hr1@192.168.56.104's password:
Permission denied, please try again.
```

This user cannot access SSH because he is not allowed to.

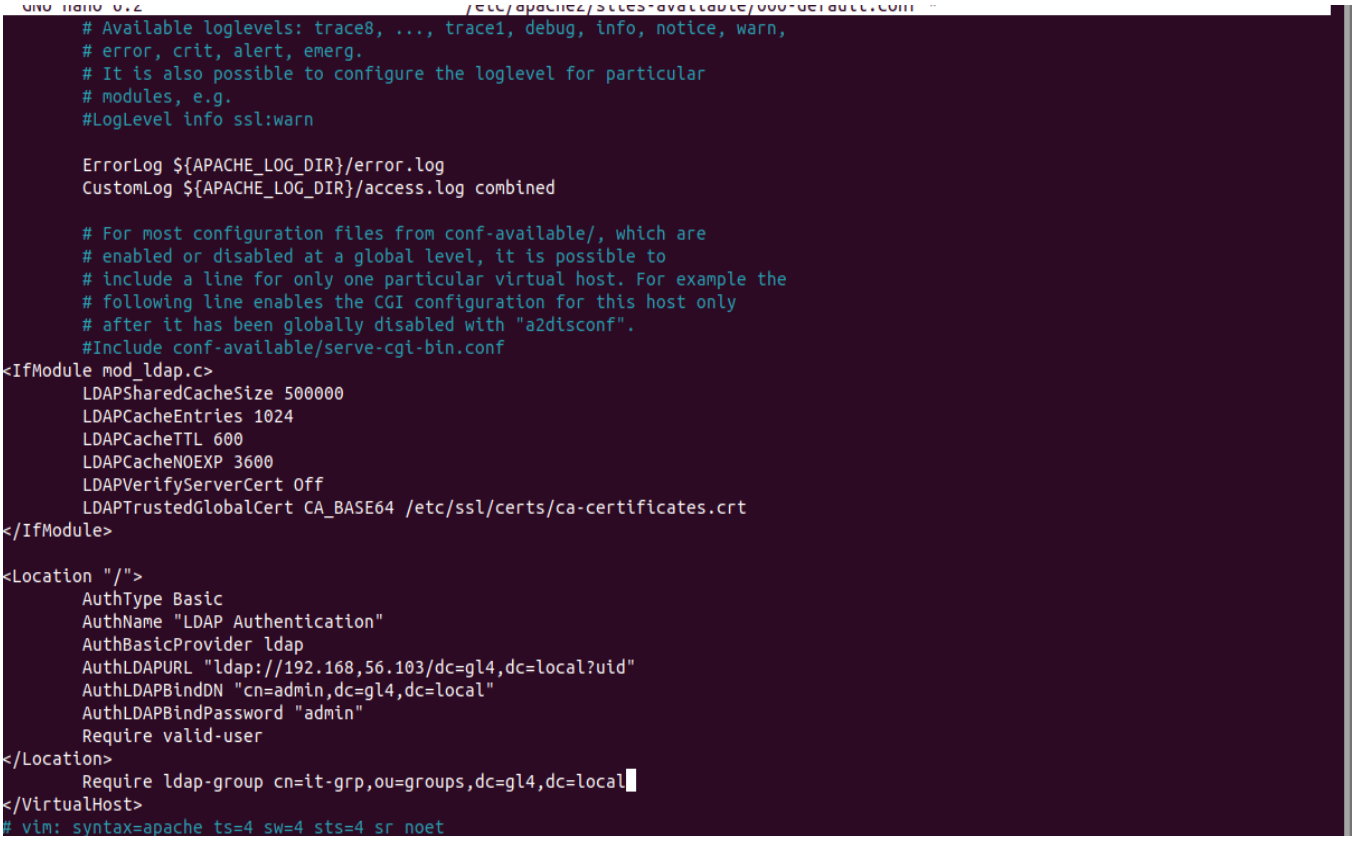
3. Integration of Apache

1. Configure Apache to use OpenLDAP authentication:

   Install apache2 on your machine : **sudo apt install apache2**

   Enable the required modules : **a2enmod ldap a2enmod authnz_ldap**

   Modify the config of the **/etc/apache2/sites-available/000-default.conf** file as below:



```
        # 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
<IfModule mod_ldap.c>
        LDAPSharedCacheSize 500000
        LDAPCacheEntries 1024
        LDAPCacheTTL 600
        LDAPCacheNOEXP 3600
        LDAPVerifyServerCert Off
        LDAPTrustedGlobalCert CA_BASE64 /etc/ssl/certs/ca-certificates.crt
</IfModule>

<Location "/">
        AuthType Basic
        AuthName "LDAP Authentication"
        AuthBasicProvider ldap
        AuthLDAPURL "ldap://192.168.56.103/dc=gl4,dc=local?uid"
        AuthLDAPBindDN "cn=admin,dc=gl4,dc=local"
        AuthLDAPBindPassword "admin"
        Require valid-user
</Location>
        Require ldap-group cn=it-grp,ou=groups,dc=gl4,dc=local
</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

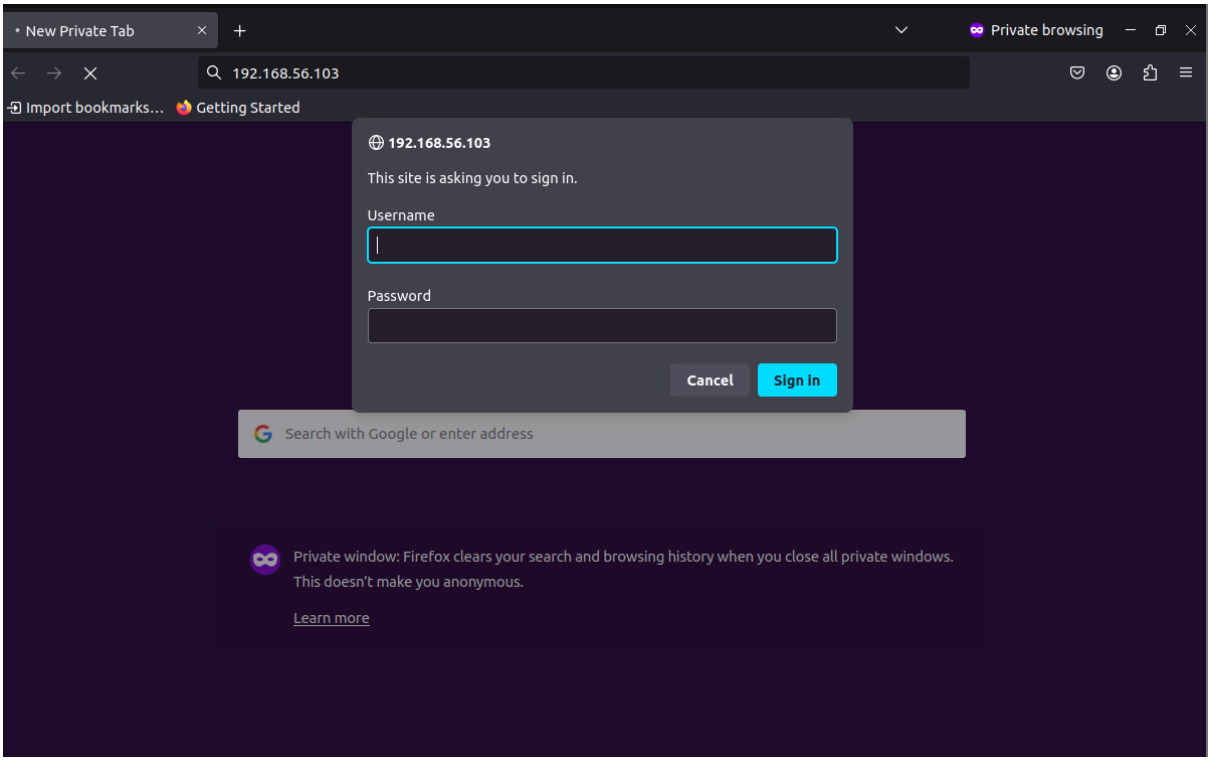
2. Restrict the access to members of a specific group:

   Add this line in the location tag of the same file to restrict the access and only allow members of “**it-grp**” to access the site

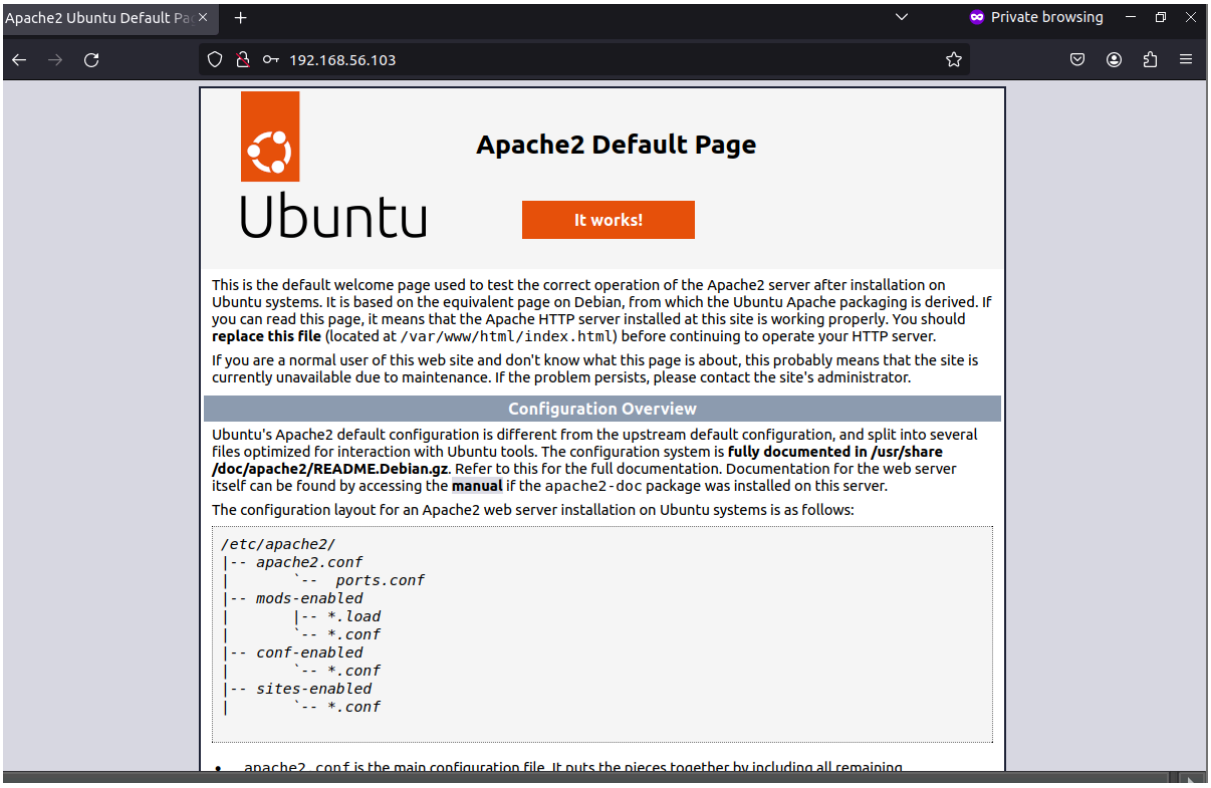
   “**Require ldap-group cn=it-grp,ou=group,dc=gl4,dc=local”**

3. Test for an authorized and an unauthorized user on a website of your choice:

   Test the access to both users **it1** and **hr1** (**it1** is allowed in and **hr1** is not allowed to access the site)



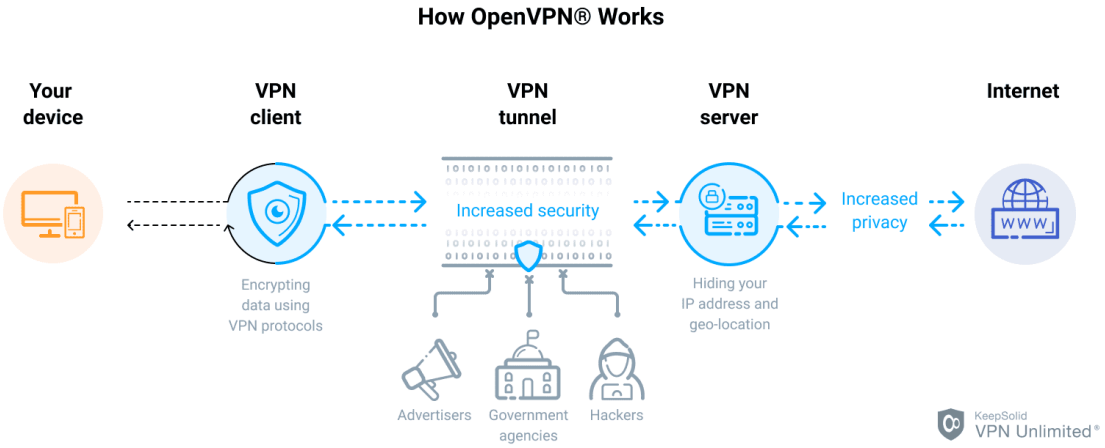
as you can see the user **it1** can access the site when he enters his credentials



4. Setting up OpenVPN

VPN or Virtual Private Network, is a technology that provides a secure and encrypted connection over a less secure network, such as the internet. It creates a "tunnel" between the user's device and a private network, allowing data to be transmitted securely as if the devices were physically connected to the private network. VPNs are commonly used to enhance security and privacy.

OpenVPN is an open-source software application that implements virtual private network (VPN) techniques to create secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities. It uses a custom security protocol that utilizes SSL/TLS for key exchange.



1. Install and configure OpenVPN to use OpenLDAP authentication:
   Server side:
   mkdir OpenVPN
   cd OpenVPN

```
tasnim@tasnim:~$ mkdir OpenVPN
tasnim@tasnim:~$ cd OpenVPN
tasnim@tasnim:~/OpenVPN$ ll
total 8
drwxrwxr-x  2 tasnim tasnim 4096 19:25 15  جانفي ./
drwxr-x--- 16 tasnim tasnim 4096 19:25 15  جانفي ../
```

   wget
   https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh

```
tasnim@tasnim:~/OpenVPN$ wget https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh
--2024-01-15 19:28:15--  https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh
```

   sudo ./openvpn-install.sh and Follow the installation steps:

```
102.157.20.7tasnim@tasnim:~/OpenVPN$ sudo ./openvpn-install.sh
Welcome to the OpenVPN installer!
The git repository is available at: https://github.com/angristan/openvpn-install

I need to ask you a few questions before starting the setup.
You can leave the default options and just press enter if you are ok with them.

I need to know the IPv4 address of the network interface you want OpenVPN listening to.
Unless your server is behind NAT, it should be your public IPv4 address.
IP address: 102.157.20.7

Checking for IPv6 connectivity...

Your host does not appear to have IPv6 connectivity.

Do you want to enable IPv6 support (NAT)? [y/n]: n

What port do you want OpenVPN to listen to?
   1) Default: 1194
   2) Custom
   3) Random [49152-65535]
Port choice [1-3]: 1

What protocol do you want OpenVPN to use?
UDP is faster. Unless it is not available, you shouldn't use TCP.
   1) UDP
   2) TCP
Protocol [1-2]: 1

What DNS resolvers do you want to use with the VPN?
   1) Current system resolvers (from /etc/resolv.conf)
   2) Self-hosted DNS Resolver (Unbound)
   3) Cloudflare (Anycast: worldwide)
   4) Quad9 (Anycast: worldwide)
   5) Quad9 uncensored (Anycast: worldwide)
   6) FDN (France)
```

install the **openvpn-auth-ldap** package using: **sudo apt-get install openvpn-auth-ldap**

```
tasnim@tasnim:~$ sudo apt install openvpn-auth-ldap
[sudo] password for tasnim:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libobjc4
```

create a file called server.conf inside the directory **/etc/openvpn/** if doesn't exist already

sudo nano /etc/openvpn/server.conf

```
 GNU nano 6.2                          /etc/openvpn/server.conf
dh /etc/openvpn/dh2048.pem
tls-server
dev tun1
ifconfig 10.0.2.1 10.0.2.2
ca /etc/ldap/sasl2/tasnim.gl4.local.crt
cert /etc/ldap/sasl2/tasnim.gl4.local.crt
key /etc/ldap/sasl2/tasnim.gl4.local.key
plugin /usr/lib/openvpn/openvpn-auth-ldap.so "/etc/openvpn/auth/ldap.conf"
```

add another file inside **/etc/openvpn/auth/ldap.conf** to configure the ldap authentication in OpenVPN

sudo nano /etc/openvpn/auth/ldap.conf

```
 GNU nano 6.2                          /etc/openvpn/auth/ldap.conf
<LDAP>
        URL ldap://192.168.56.103:389
        BindDN cn=admin,dc=gl4,dc=local
        Password admin
        Timeout 15
        TLSEnable no
        FollowReferrals yes
</LDAP>

<Authorization>
        BaseDN "ou=People,dc=gl4,dc=local"
        SearchFilter "(uid=%u)"
        RequireGroup false
</Authorization>
```

Client side:

sudo apt update

```
client@client:~$ sudo apt update
[sudo] password for client:
```

sudo apt install openssh-client openssh-server -y

```
client@client:~$ sudo apt install openssh-client openssh-server -y
Reading package lists... Done
```

Create a **Client.conf** file inside **/etc/openvpn/**

sudo nano /etc/openvpn/client.conf

```
 GNU nano 6.2                          /etc/openvpn/client.conf
tls-client
dev tun1
ifconfig 10.0.2.2 10.0.2.1
remote 192.168.56.104
ca /etc/ldap/sasl2/tasnim.gl4.local.crt
cert /etc/ldap/sasl2/tasnim.gl4.local.crt
key /etc/ldap/sasl2/tasnim.gl4.local.key
pull
auth-user-pass
```

2. Test the VPN connection using OpenLDAP:

   Test the vpn connection using openLDAP authentication :

   sudo openvpn –config /etc/openvpn/server.conf

```
client@client:/etc/pam.d$ sudo openvpn --config /etc/openvpn/server.conf
2024-01-18 00:29:46 Cipher negotiation is disabled since neither P2MP client nor server mode is enabled
2024-01-18 00:29:46 OpenVPN 2.5.9 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Sep
29 2023
2024-01-18 00:29:46 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
2024-01-18 00:29:46 WARNING: INSECURE cipher (BF-CBC) with block size less than 128 bit (64 bit).  This allows attacks like SWEET32.
  Mitigate by using a --cipher with a larger block size (e.g. AES-256-CBC). Support for these insecure ciphers will be removed in Op
enVPN 2.7.
2024-01-18 00:29:46 TUN/TAP device tun1 opened
2024-01-18 00:29:46 net_iface_mtu_set: mtu 1500 for tun1
2024-01-18 00:29:46 net_iface_up: set tun1 up
2024-01-18 00:29:46 net_addr_ptp_v4_add: 10.0.2.1 peer 10.0.2.2 dev tun1
2024-01-18 00:29:46 Could not determine IPv4/IPv6 protocol. Using AF_INET
2024-01-18 00:29:46 UDPv4 link local (bound): [AF_INET][undef]:1194
2024-01-18 00:29:46 UDPv4 link remote: [AF_UNSPEC]
2024-01-18 00:29:54 peer info: IV_VER=2.5.9
2024-01-18 00:29:54 peer info: IV_PLAT=linux
2024-01-18 00:29:54 peer info: IV_PROTO=6
2024-01-18 00:29:54 peer info: IV_NCP=2
2024-01-18 00:29:54 peer info: IV_CIPHERS=AES-256-GCM:AES-128-GCM
2024-01-18 00:29:54 peer info: IV_LZ4=1
2024-01-18 00:29:54 peer info: IV_LZ4v2=1
2024-01-18 00:29:54 peer info: IV_LZO=1
2024-01-18 00:29:54 peer info: IV_COMP_STUB=1
2024-01-18 00:29:54 peer info: IV_COMP_STUBv2=1
2024-01-18 00:29:54 peer info: IV_TCPNL=1
2024-01-18 00:29:54 WARNING: 'ifconfig' is present in local config but missing in remote config, local='ifconfig 10.0.2.1 10.0.2.2'
2024-01-18 00:29:54 WARNING: INSECURE cipher (BF-CBC) with block size less than 128 bit (64 bit).  This allows attacks like SWEET32.
  Mitigate by using a --cipher with a larger block size (e.g. AES-256-CBC). Support for these insecure ciphers will be removed in Op
enVPN 2.7.
2024-01-18 00:29:54 WARNING: INSECURE cipher (BF-CBC) with block size less than 128 bit (64 bit).  This allows attacks like SWEET32.
  Mitigate by using a --cipher with a larger block size (e.g. AES-256-CBC). Support for these insecure ciphers will be removed in Op
enVPN 2.7.
2024-01-18 00:29:54 WARNING: cipher with small block size in use, reducing reneg-bytes to 64MB to mitigate SWEET32 attacks.
2024-01-18 00:29:54 [tasnim.gl4.local] Peer Connection Initiated with [AF_INET]192.168.56.103:1194
2024-01-18 00:29:55 Initialization Sequence Completed
2024-01-18 00:30:00 peer info: IV_VER=2.5.9
```

sudo openvpn –config /etc/openvpn/client.conf

```
tasnim@tasnim:~$ sudo systemctl start openvpn
tasnim@tasnim:~$ sudo openvpn --config /etc/openvpn/client.conf
2024-01-18 00:29:48 --cipher is not set. Previous OpenVPN version defaulted to BF-CBC as fallback w
hen cipher negotiation failed in this case. If you need this fallback please add '--data-ciphers-fa
llback BF-CBC' to your configuration and/or add BF-CBC to --data-ciphers.
2024-01-18 00:29:48 OpenVPN 2.5.9 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11]
[MH/PKTINFO] [AEAD] built on Sep 29 2023
2024-01-18 00:29:48 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
Enter Auth Username: it1
🔐 Enter Auth Password: *****
2024-01-18 00:29:54 WARNING: using --pull/--client and --ifconfig together is probably not what you
 want
2024-01-18 00:29:54 WARNING: No server certificate verification method has been enabled.  See http:
//openvpn.net/howto.html#mitm for more info.
2024-01-18 00:29:54 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.56.104:1194
```

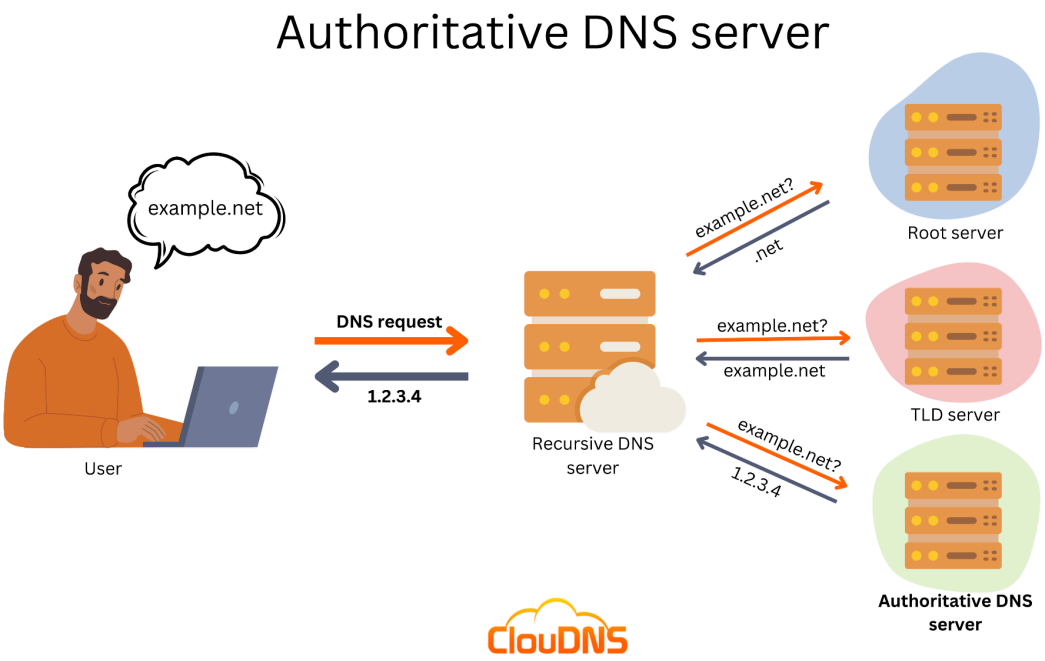
3. Test for an authorized client and an unauthorized client:

Test the connection for “**it1**” a user who is allowed in and an “**hr1**” user who is not allowed:

```
tasnim@tasnim:~$ sudo systemctl start openvpn
tasnim@tasnim:~$ sudo openvpn --config /etc/openvpn/client.conf
2024-01-18 00:29:48 --cipher is not set. Previous OpenVPN version defaulted to BF-CBC as fallback w
hen cipher negotiation failed in this case. If you need this fallback please add '--data-ciphers-fa
llback BF-CBC' to your configuration and/or add BF-CBC to --data-ciphers.
2024-01-18 00:29:48 OpenVPN 2.5.9 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11]
[MH/PKTINFO] [AEAD] built on Sep 29 2023
2024-01-18 00:29:48 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
Enter Auth Username: it1
🔐 Enter Auth Password: *****
2024-01-18 00:29:54 WARNING: using --pull/--client and --ifconfig together is probably not what you
 want
2024-01-18 00:29:54 WARNING: No server certificate verification method has been enabled.  See http:
//openvpn.net/howto.html#mitm for more info.
2024-01-18 00:29:54 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.56.104:1194
```

Network Services Management with DNS

DNS (Domain Name System) is a system that translates human-readable domain names (like www.example.com) into numerical IP addresses that computers use to identify each other on the Internet. It's like a phonebook for the Internet, helping your device find the right destination when you enter a website name.



1. DNS Server Configuration on a separate machine

Step 1 : Add the IP and FQDN to file **/etc/hosts**

```
tasnim@tasnim:/etc/bind$ sudo cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       tasnim.security.local tasnim
192.168.56.105  tasnim.security.local tasnim

# 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
```

You can verify changes using the following commands :

```
tasnim@tasnim:/etc/bind$ sudo nano /etc/hosts
tasnim@tasnim:/etc/bind$ hostname
tasnim
tasnim@tasnim:/etc/bind$ dnsdomainname
security.local
tasnim@tasnim:/etc/bind$ hostname -f
tasnim.security.local
```

Step 2 : Install DNS package : **sudo apt-get install bind9**

Step 3 : Edit file **/etc/bind/named.conf.options** , (make sure to create a backup copy before modifying it)

```
tasnim@tasnim:/etc/bind$ cat named.conf.options
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 {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        listen-on-v6 { any; };
        recursion yes;
        listen-on {192.168.56.105;};
        allow-transfer {none;};

#       forwarders {
#       192.168.56.105;
#       };
};
```

Step 4 : Edit file **/etc/bind/named.conf.local** , (make sure to create a backup copy before modifying it)

```
tasnim@tasnim:/etc/bind$ cat named.conf.local
//
// 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 "security.local" IN {
        type master;
        file "/etc/bind/db.security.local";
};

//reverse lookup zone
zone "56.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/nind/db.56.168.192";
};
```

Step 4 : Create file file **/etc/bind/db.security.local** (add the necessary DNS records for the OpenLDAP, Apache, and OpenVPN servers)

```
tasnim@tasnim:/etc/bind$ sudo cat db.security.local
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.security.local. root.security.local. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.security.local.
ns1     IN      A       192.168.56.105
client  IN      A       192.168.56.104
tasnim  IN      A       192.168.56.103
@       IN      A       192.168.56.105
```

Step 4 : Create file file **/etc/bind/db.56.168.192** (reverse)

```
tasnim@tasnim:/etc/bind$ cat db.56.168.192
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.security.local root.security.local. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.
105     IN      PTR     ns1.security.local.
```

Step 5 : Edit file **/etc/resolv.conf**

```
nameserver 192.168.56.105
domaine security.local
search security.local
```

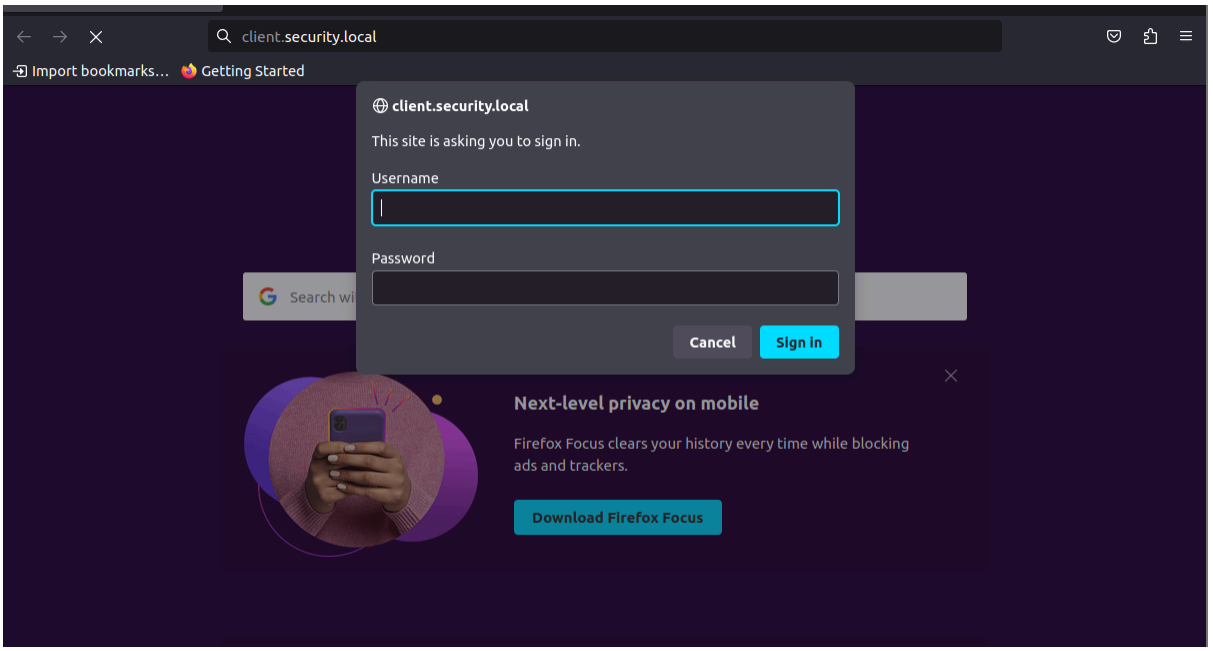
Step 6: Restart the bind9 server : **sudo service bind9 restart**

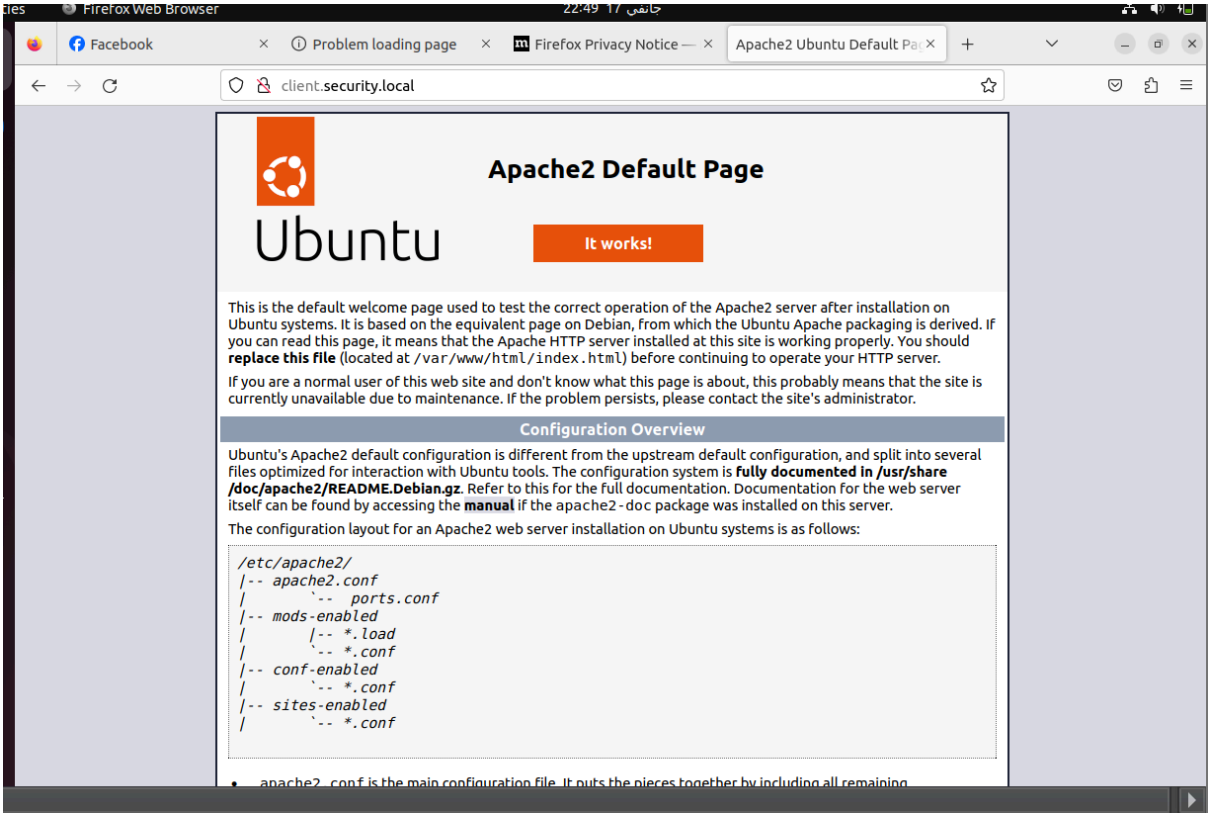
```
tasnim@tasnim:/etc/bind$ sudo service bind9 status
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-01-17 22:36:32 CET; 15s ago
       Docs: man:named(8)
    Process: 3878 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 3881 (named)
      Tasks: 3 (limit: 2261)
     Memory: 6.5M
        CPU: 123ms
     CGroup: /system.slice/named.service
             └─3881 /usr/sbin/named -u bind

22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1084.awsdns-07.org/A/IN': 2600:9000:5302:c900::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1084.awsdns-07.org/AAAA/IN': 2600:9000:5304:a00::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1084.awsdns-07.org/A/IN': 2600:9000:5304:a00::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1084.awsdns-07.org/AAAA/IN': 2600:9000:5306:4700::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1084.awsdns-07.org/A/IN': 2600:9000:5306:4700::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1260.awsdns-29.org/AAAA/IN': 2600:9000:5302:df00::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1260.awsdns-29.org/AAAA/IN': 2600:9000:5304:2000::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1260.awsdns-29.org/AAAA/IN': 2600:9000:5300:9d00::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1260.awsdns-29.org/AAAA/IN': 2600:9000:5306:5d00::1#53
22:36:38 17 جانفي tasnim named[3881]: network unreachable resolving 'ns-1260.awsdns-29.org/A/IN': 2600:9000:5300:9d00::1#53
tasnim@tasnim:/etc/bind$
```

2. Validation and Testing

```
tasnim@tasnim:/etc/bind$ sudo service bind9 restart
tasnim@tasnim:/etc/bind$ nslookup main.security.local
Server:         192.168.56.105
Address:        192.168.56.105#53

Name:   main.security.local
Address: 192.168.56.105

tasnim@tasnim:/etc/bind$ nslookup tasnim.security.local
Server:         192.168.56.105
Address:        192.168.56.105#53

Name:   tasnim.security.local
Address: 192.168.56.103

tasnim@tasnim:/etc/bind$ nslookup client.security.local
Server:         192.168.56.105
Address:        192.168.56.105#53

Name:   client.security.local
Address: 192.168.56.104
```

Testing:


client.security.local
Import bookmarks...
Getting Started
client.security.local
This site is asking you to sign in.
Username
Password
Cancel
Sign in
Next-level privacy on mobile
Firefox Focus clears your history every time while blocking ads and trackers.
Download Firefox Focus


Firefox Web Browser
Facebook
Problem loading page
Firefox Privacy Notice
Apache2 Ubuntu Default Pa
client.security.local
Apache2 Default Page
Ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf

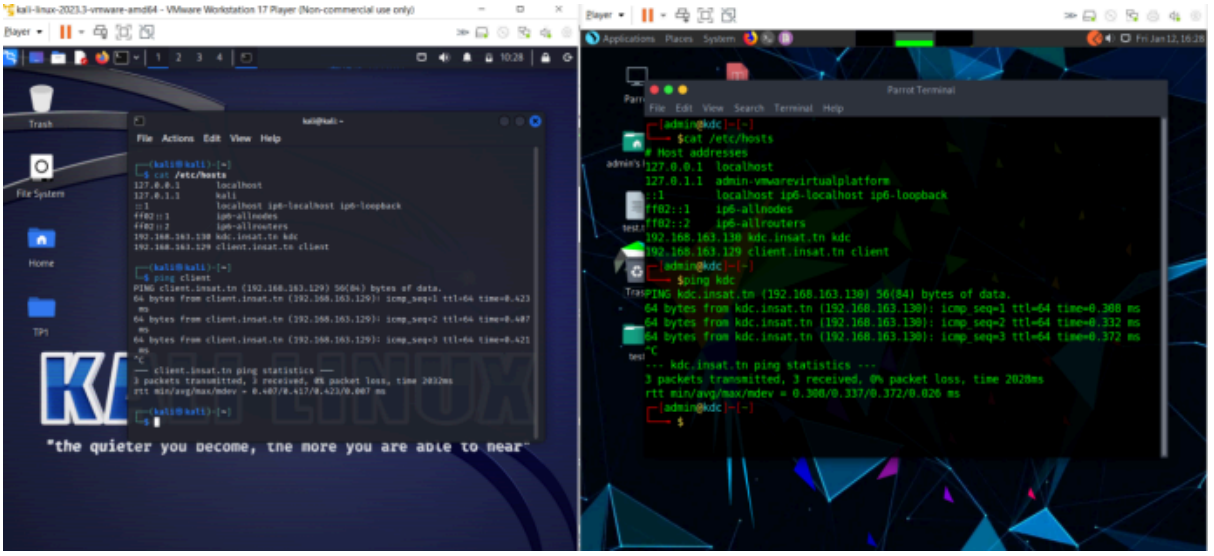
Authentication with Kerberos

1. Kerberos Server Configuration

1.1 Initialisation & Server Configuration

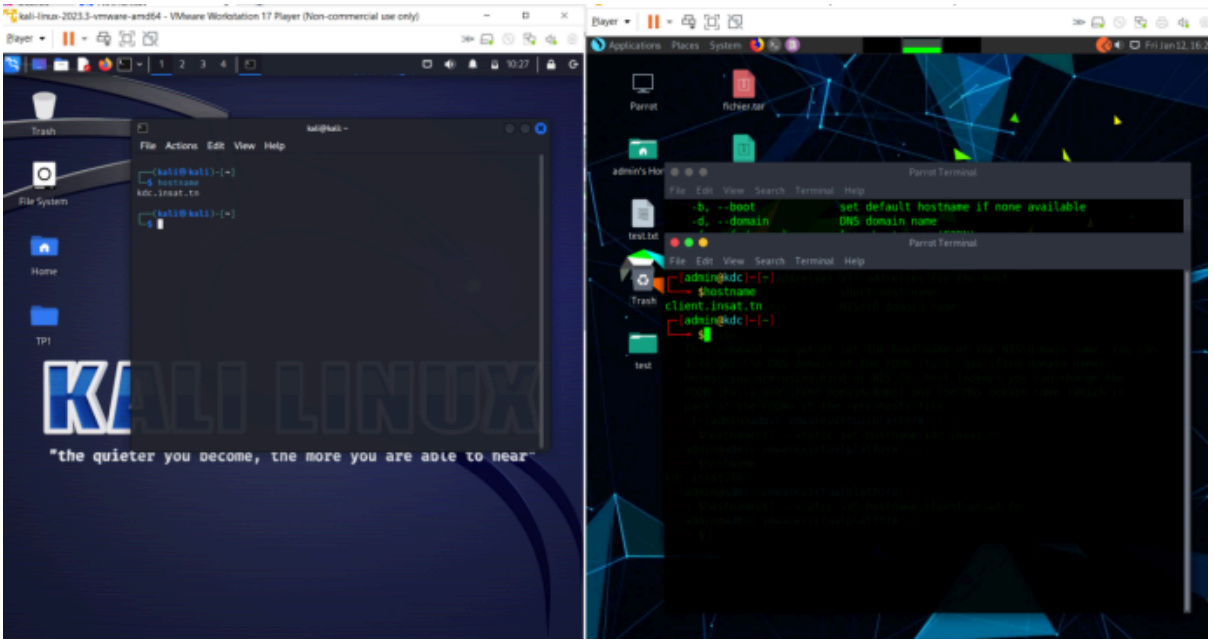
We started by configuring our DNS and modifying the client's and server's machines hostnames

sudo nano /etc/hosts to edit, then cat to show the file

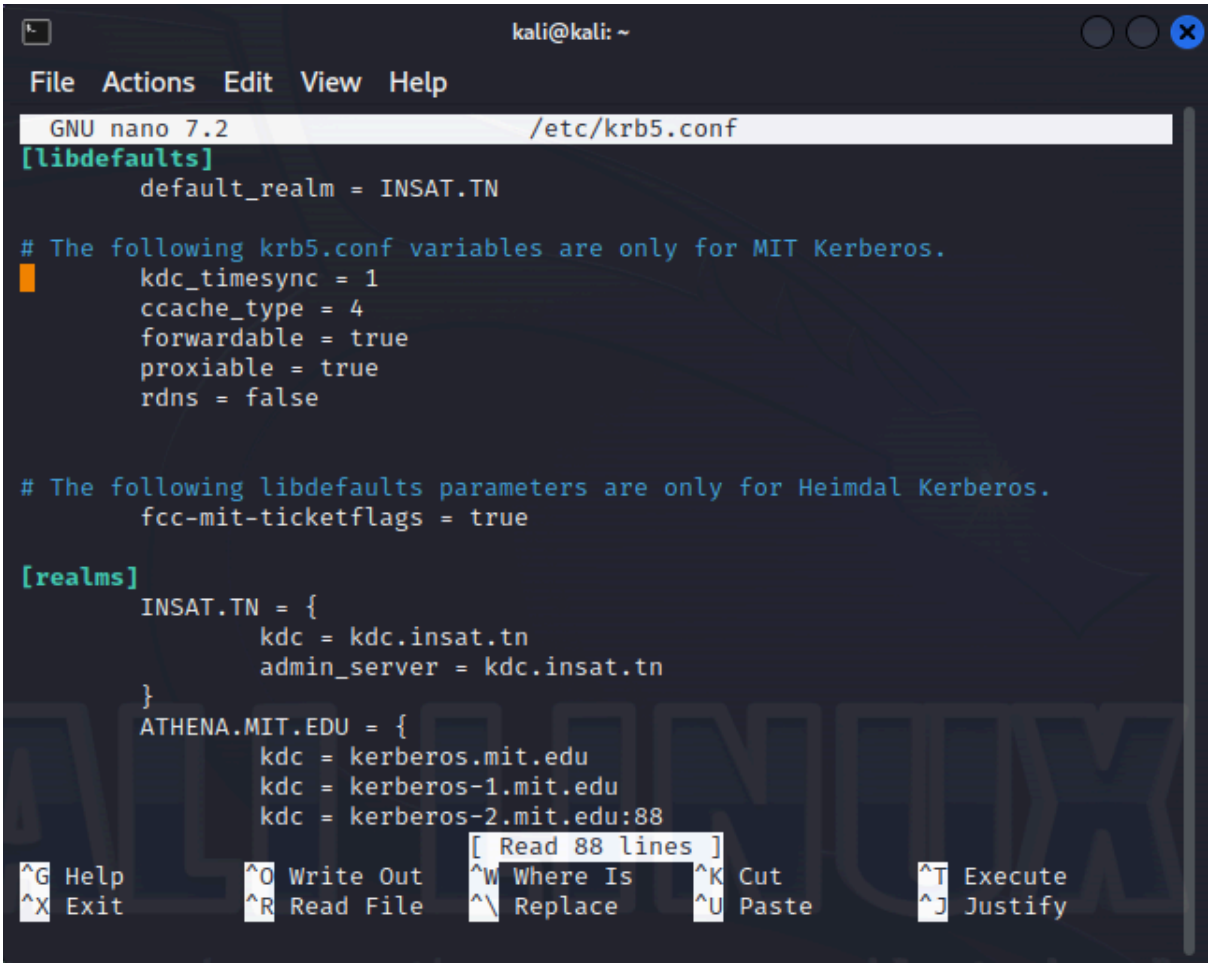


we can then ping our two machines using **ping client** and **ping kdc**

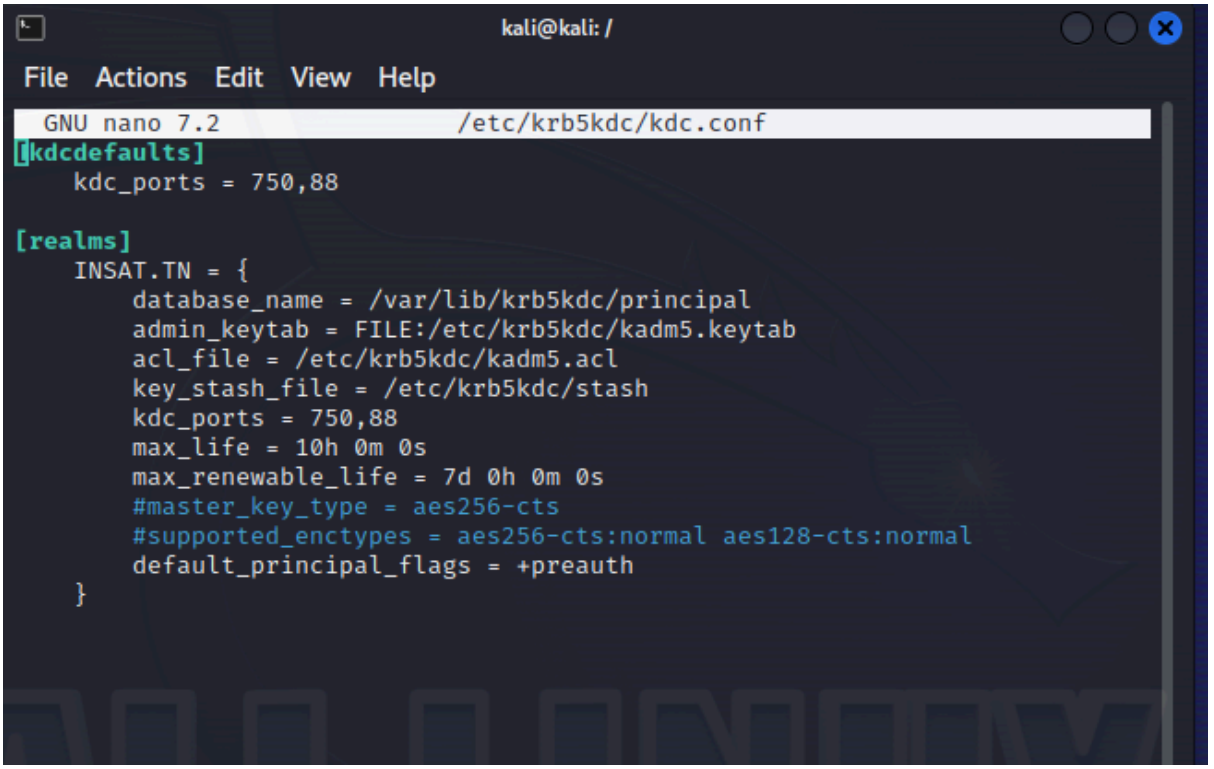
sudo hostnamectl set-hostname kdc.insat.tn (on the server)
sudo hostnamectl set-hostname client.insat.tn (on the client)



Once that done, we installed the Kerberos packages and set our default realm to be INSAT.TN, as stated in the krb5.conf file:

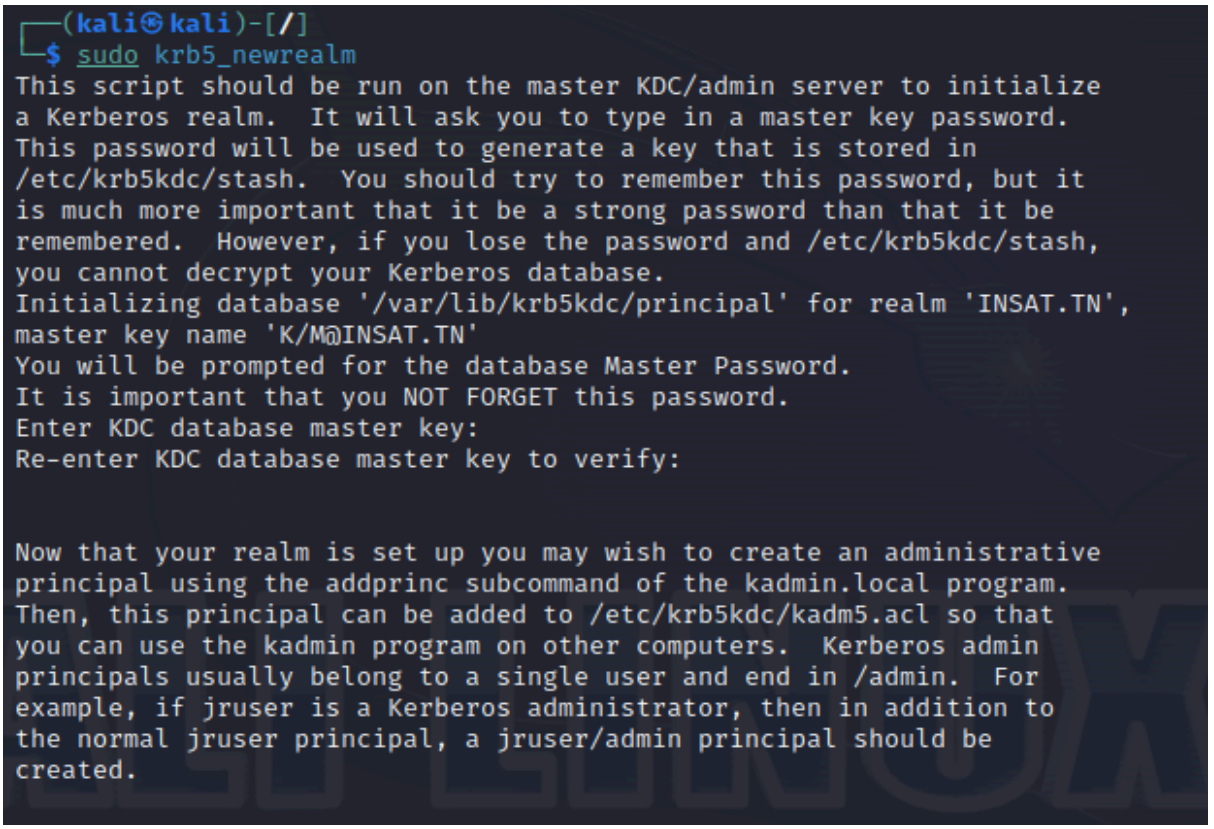


For upcoming steps, we will need some other infos about our Kerberos setup such as the location of the keytab. For that, we can check the /etc/krb5kdc/kdc.conf file:



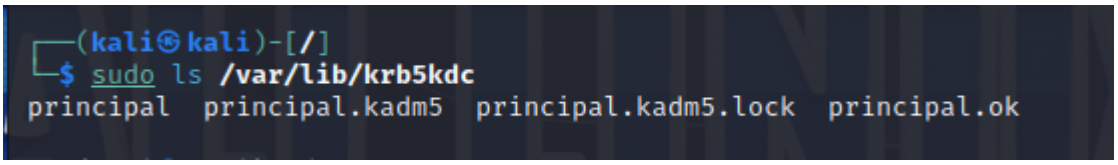
We will then execute the krb5_newrealm command and set a password for our database (KDC database master key):

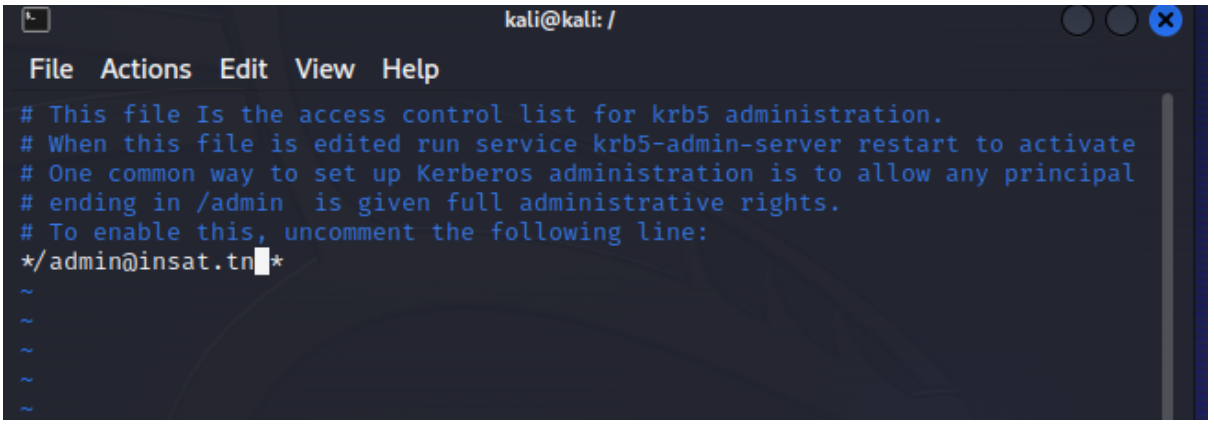
sudo krb5_newrealm



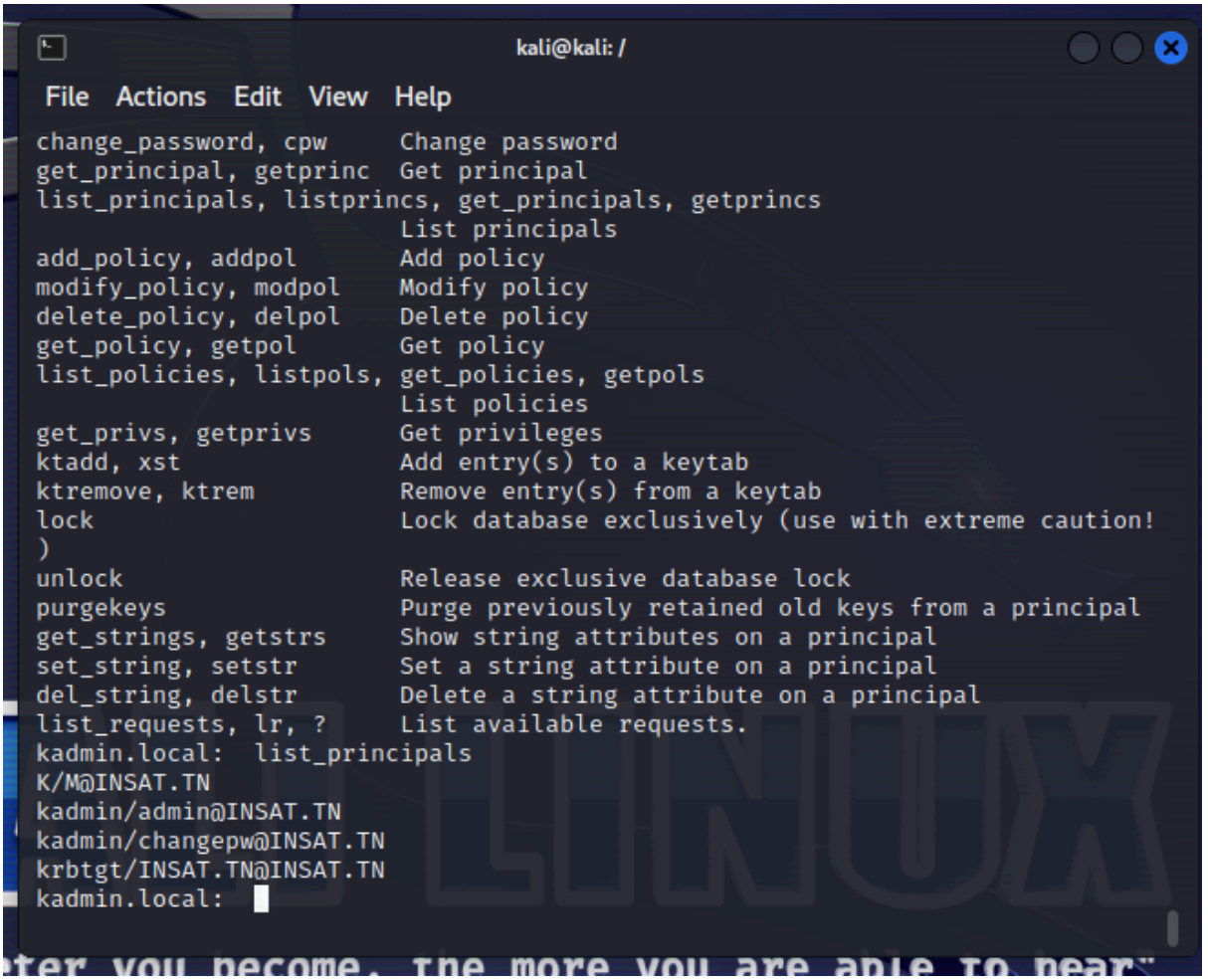
We can then ensure that the process was completed successfully by checking the new files generated under /var/lib/krb5kdc

ls /var/lib/krb5kdc





We can then execute Kadmin.locale to execute various commands related to kerberos, such as listing principals and creating new ones



We will add the principal: “utilisateur” which we will later on use to login via ssh

add_principal utilisateur

```
kadmin.local:  add_principal utilisateur
No policy specified for utilisateur@INSAT.TN; defaulting to no policy
Enter password for principal "utilisateur@INSAT.TN":
Re-enter password for principal "utilisateur@INSAT.TN":
Principal "utilisateur@INSAT.TN" created.
kadmin.local:  list_principals
K/M@INSAT.TN
kadmin/admin@INSAT.TN
kadmin/changepw@INSAT.TN
krbtgt/INSAT.TN@INSAT.TN
utilisateur@INSAT.TN
kadmin.local:
```

We then add an admin principal

add_principal root/admin

```
kadmin.local:  add_principal 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:  list_principals
K/M@INSAT.TN
kadmin/admin@INSAT.TN
kadmin/changepw@INSAT.TN
krbtgt/INSAT.TN@INSAT.TN
root/admin@INSAT.TN
utilisateur@INSAT.TN
kadmin.local:
```

We add a host

add_principal host/kdc.insat.tn

```
┌──(kali㉿kali)-[/]
└─$ sudo /usr/sbin/kadmin.local
Authenticating as principal root/admin@INSAT.TN with password.
kadmin.local:  add_principal host/kdc.insat.tn
No policy specified for host/kdc.insat.tn@INSAT.TN; defaulting to no policy
Enter password for principal "host/kdc.insat.tn@INSAT.TN":
Re-enter password for principal "host/kdc.insat.tn@INSAT.TN":
Principal "host/kdc.insat.tn@INSAT.TN" created.
kadmin.local:
```

Once all of that is ready, we can run kinit to generate a ticket for the root/admin principal. We ensure that the ticket was generated by running:

klist

```
┌──(kali㉿kali)-[/]
└─$ kinit -p root/admin
Password for root/admin@INSAT.TN:

┌──(kali㉿kali)-[/]
└─$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: root/admin@INSAT.TN

Valid starting       Expires              Service principal
01/12/2024 10:50:29  01/12/2024 20:50:29  krbtgt/INSAT.TN@INSAT.TN
        renew until 01/13/2024 10:50:27
```

We will then add entries to the keytab for the admin and the host

```
┌──(kali㉿kali)-[/etc]
└─$ sudo file /etc/krb5kdc/kadm5.keytab
/etc/krb5kdc/kadm5.keytab: Kerberos Keytab file, realm=INSAT.TN, principal=ro
ot/admin, type=1, date=Fri Jan 12 16:04:54 2024, kvno=1
```

```
┌──(kali㉿kali)-[/etc]
└─$ sudo /usr/bin/ktutil
ktutil:  addent -password -p host/kdc.insat.tn -k 1 -e aes256-cts-hmac-sha1-9
6
Password for host/kdc.insat.tn@INSAT.TN:
ktutil:
```

Once that done, we can add a password policy for our principals:

add_policy -minlength 3 -maxlife 90d “policy1”

```
kadmin.local:  add_policy -minlength 3 -maxlife 90d "policy1"
kadmin.local:  modify_principal -policy "policy1" root/admin
Principal "root/admin@INSAT.TN" modified.
kadmin.local:  modify_principal -policy "policy1" host/kdc.insat.tn
Principal "host/kdc.insat.tn@INSAT.TN" modified.
kadmin.local:
```

In here, we set the minimum length of the password to 3 (for easy testing purposes) and the maxlife of the password to 90 days.

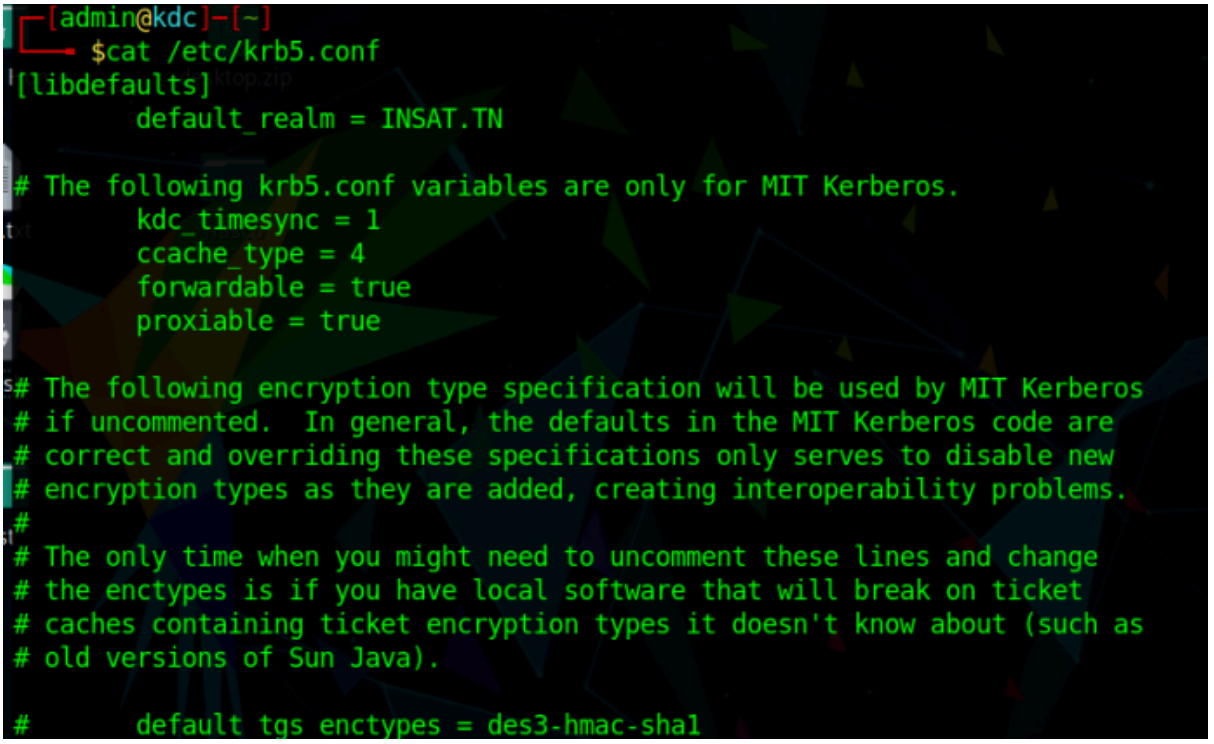
2. Authentication with SSH

The selection of SSH for Kerberos authentication is justified by its ease of integration and its widespread use in remote system management. SSH, commonly employed for its robust security features, including encrypted communications, pairs seamlessly with

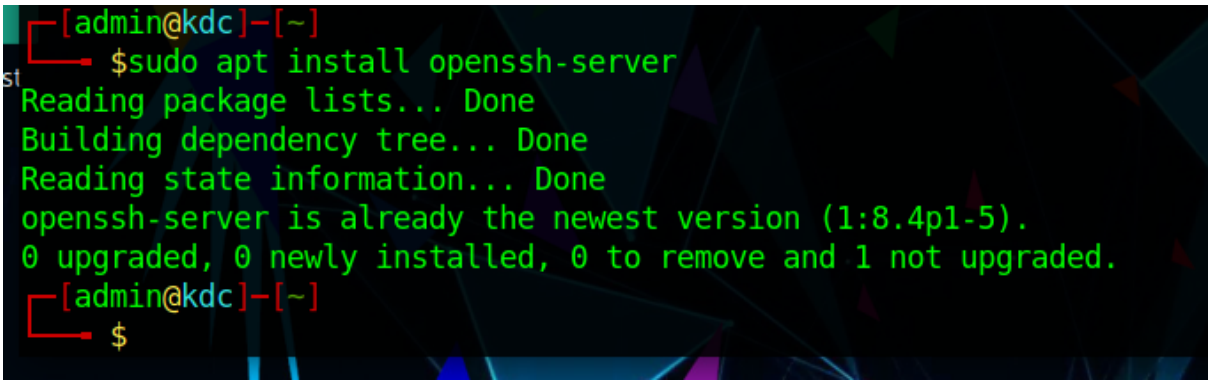
Kerberos for centralized and secure authentication. This combination enhances access security while remaining practical and relevant in today's computing environment.

Our first step here will be to configure kerberos in the client's machine by intalling the kerberos packages, and providing the same default realm (INSAT.TN).

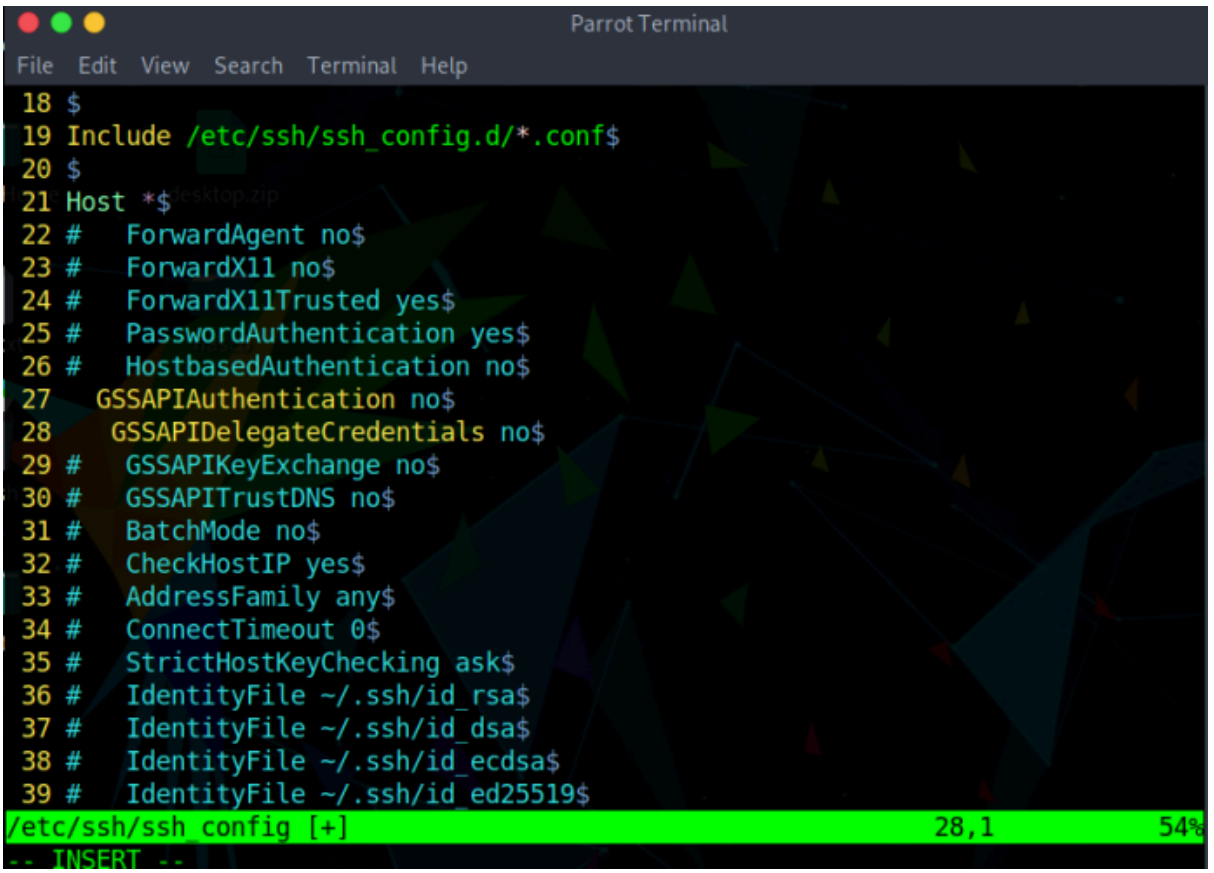
The main purpose of this step is to have the same krb5.conf in both machines. **We could also scp the krb5.conf file from the server to the client.**



Once that done, we install openssh-server

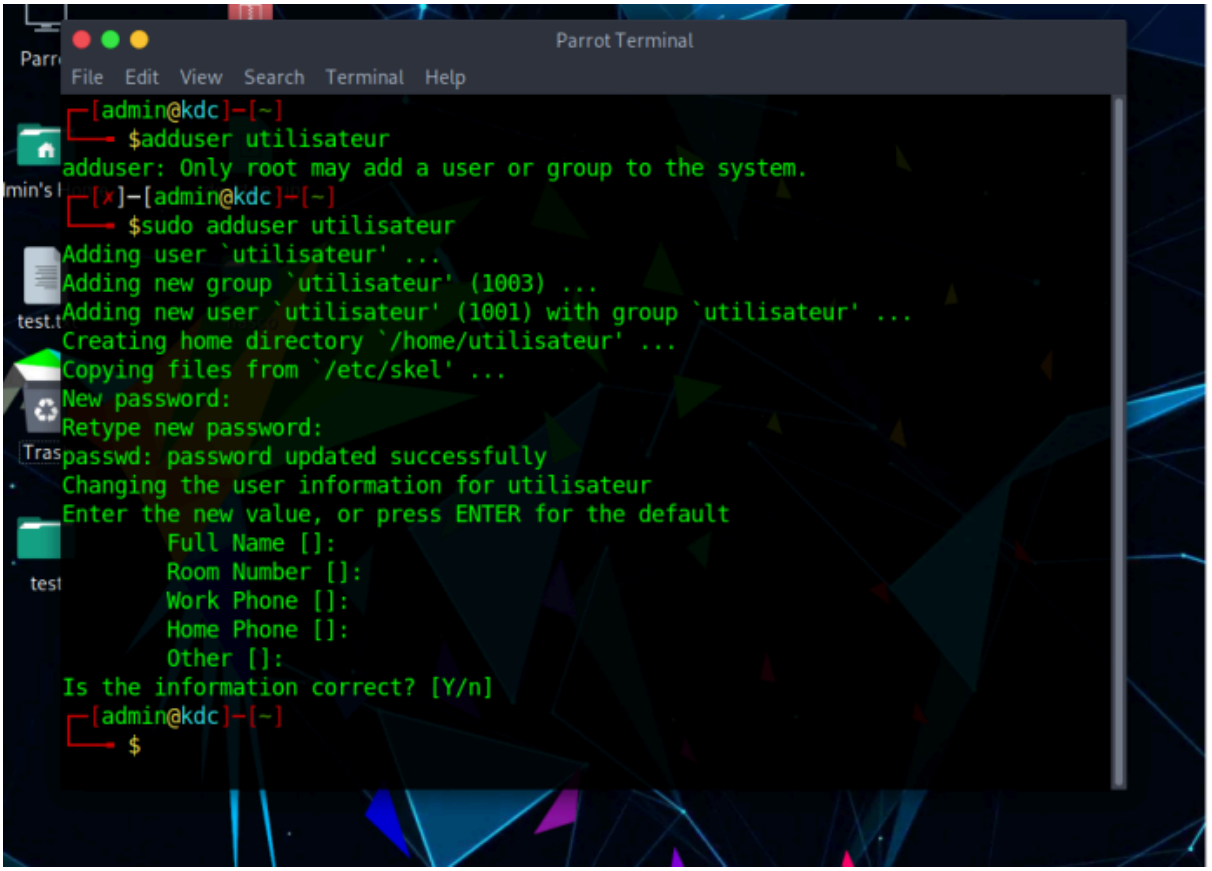


We enable the authentication via GSAPI (we also do this step in the server's sshd config)



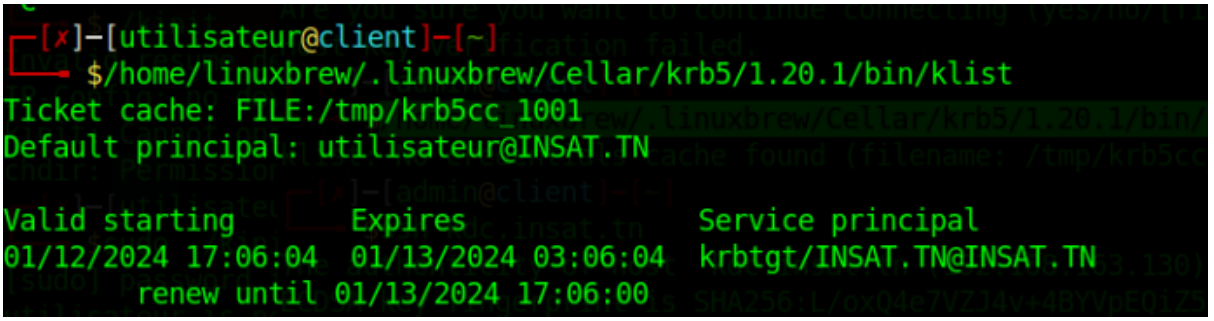
In order to faciliate the login process, we will create a new user, ‘utilisateur’, named after the principal we have in the Kerberos server

adduser utilisateur



Once that done, if we try to ssh to the server, it will ask for a password!

To use kerberos instead, we will simply need to generate a ticket with **kinit**. Once that done, we can make sure we obtained the ticket using **klist**

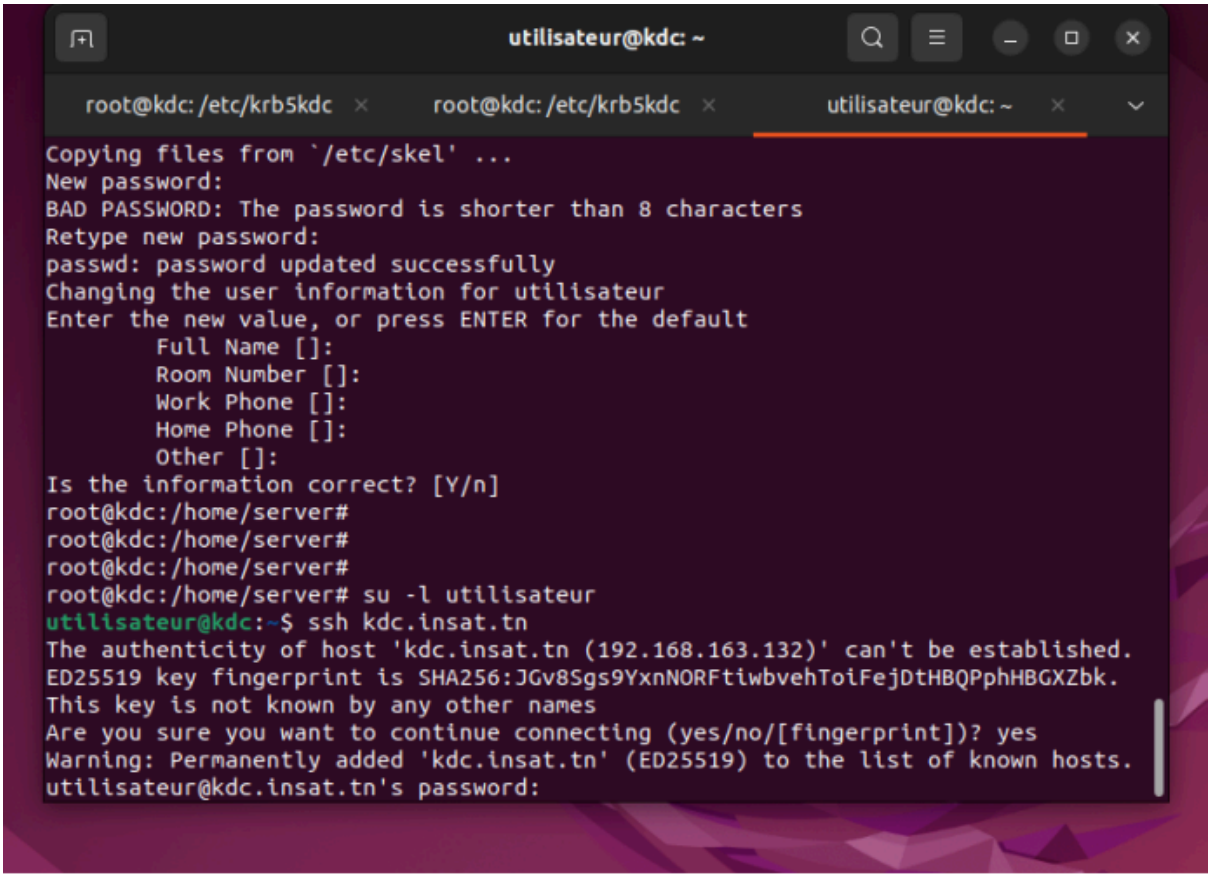


Important note:

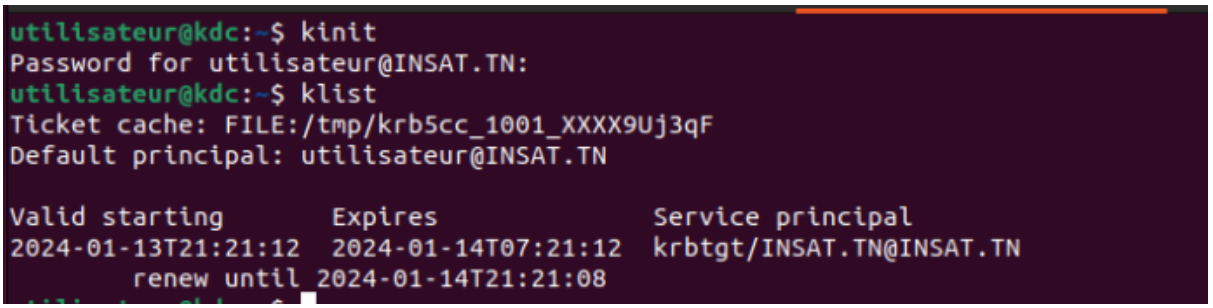
=> At this point, we had a valid ticket, a properly configured ssh server and ssh client, and kerberos was properly setup. Unfortunately, after trying multiple times (we spent more than 12 hours debugging this issue), it was impossible to get ssh to login to the server without providing a password. We inspected ssh's logs and found that the «connection» methods used included «gssapi-with-mic» which is the method related to kerberos, but it was failing for an unmentionned reason (it stated that the server was returning a «packet: type 51» which isn't a valid answer.

After reading about this issue, we concluded that using Kali was the cause of the problem so we ended up switching to an Ubuntu machine and repeating the exact same steps, and it worked properly!

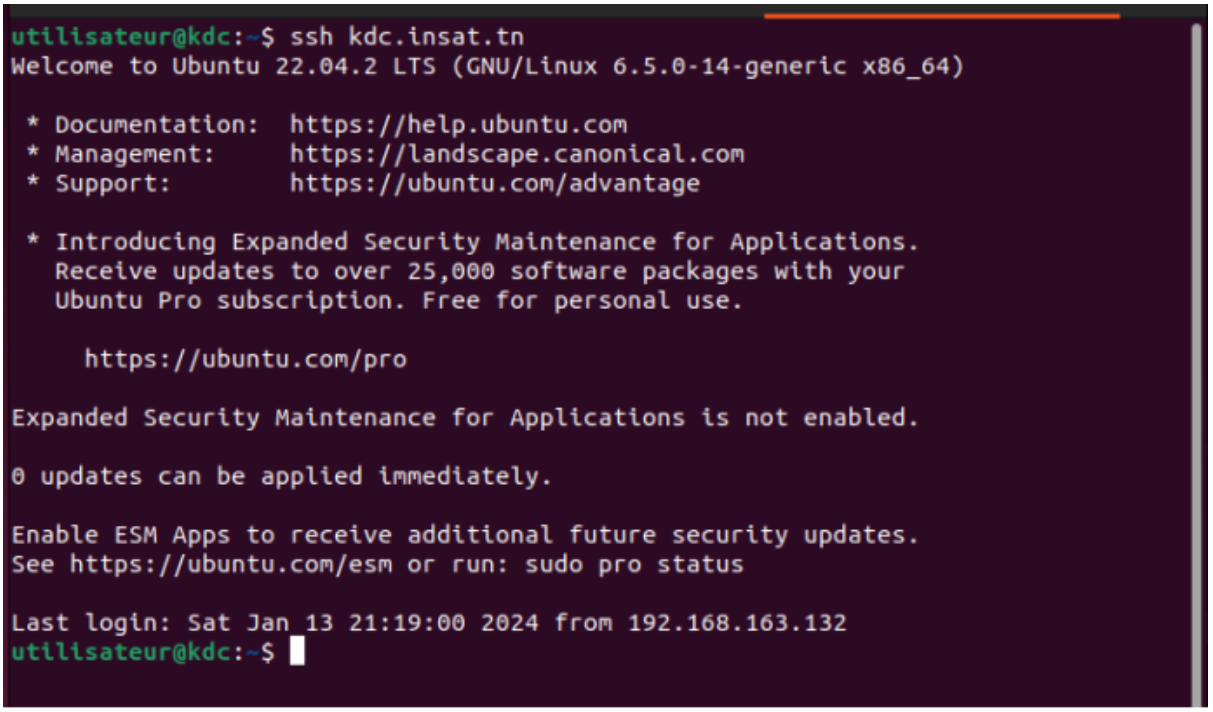
Here's the first SSH where we were asked for a password (before generating a ticket)



We then generated a ticket using **kinit** and displayed it with **klist**:



And finally, we used SSH again and kerberos was being properly used, as we weren't asked for a password!


utilisateur@kdc:~$ ssh kdc.insat.tn
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 6.5.0-14-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
* Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.
https://ubuntu.com/pro
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
Last login: Sat Jan 13 21:19:00 2024 from 192.168.163.132
utilisateur@kdc:~$