

Practica 3 - Servidor NFS y LDAP

Hayk Kocharyan

April 16, 2020

Para el desarrollo de esta práctica se exigía la implementación de dos funcionalidades principales en una máquina que corre el sistema operativo Ubuntu en su versión 16.04.

- Servidor NFS, para acceso remoto a un sistema de archivos a través de la red.
- Servidor LDAP, usado como BBDD que ofrece un servicio de directorios y usuarios distribuido.

El diagrama del sistema es el siguiente:

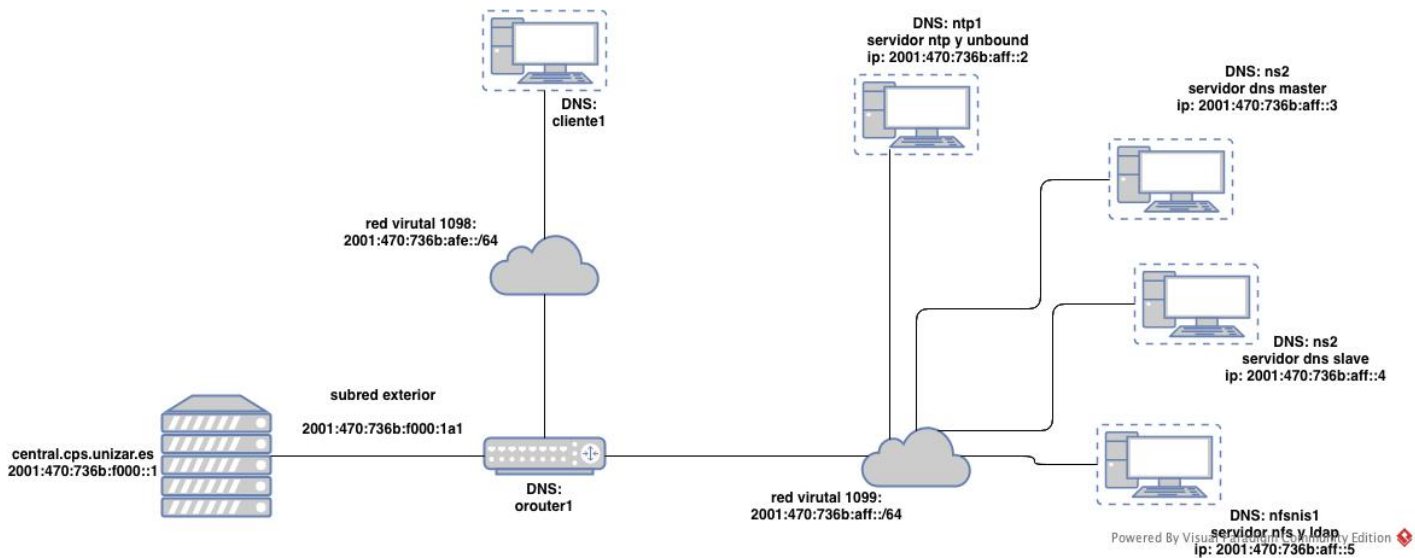


Figure 1: Despliegue del sistema

Como novedad en esta práctica tenemos:

- La máquina **nfsnis1** que será la encargada de ofrecer un servicio de sistemas de ficheros en red para sus clientes, y además también ofrecerá un servicio de cuentas de usuario a través de **LDAP**.
- La máquina **cliente1** que se encuentra en una nueva subred virtual, esta será la máquina de pruebas para el servidor NFS y LDAP.

Las primeras configuraciones que se han realizado en las dos nuevas máquinas han consistido en:

- añadir **las nuevas interfaces** de red en las máquinas ubuntu (cliente1 y nfsnis1).
- Añadir ambas máquinas al fichero de zonas del servidor dns maestro.
- Añadir el servicio ntp, modificando el fichero `/etc/ntp.conf` dejando únicamente la dirección de nuestro servidor ntp.
- Configurar el servicio dns dejando en el fichero `/etc/resolvconf/resolv.conf.d/head` la entrada: **nameserver @unbound** y reiniciar el demonio para aplicar los cambios a resolv.conf.

1 NFS

1.1 Máquina servidor - nfsnis1

Para configurar NFS en el servidor, en primer lugar hemos creado un directorio en `/home` llamado **ldapUsers** donde ubicaremos los directorios home de cada usuario ldap para ser exportado con mas comodidad. No he optado por exportar `/home` ya que de esta manera daríamos acceso a los clientes a todo el directorio, cosa que puede no ser muy segura. A continuación, realizamos un bind del directorio a exportar con `/srv/nfs4/home`, nuestro directorio de exportación. Finalmente, añadiremos el último directorio mencionado al fichero de exports y reiniciaremos el servicio. Podemos ver los pasos detallados en el anexo 1, en la sección de servidor nfs4.

1.2 Máquina cliente - cliente1

En el cliente simplemente tenemos que montar el directorio que nos exporta el servidor, y pasamos a tener acceso completo a este. Los pasos detallados se encuentran en el anexo uno en la sección de cliente NFS.

2 LDAP

2.1 Máquina servidor - nfsnis1

En primer lugar instalamos *slapd*, el demonio que escuchará las peticiones LDAP, y establecemos la contraseña del administrador, y *ldap-utils* (nos ofrece las herramientas para modificar las entradas). A continuación, instalaremos *libnss-ldap*, este paquete nos provee un servicio NSS que permite al servidor LDAP actuar como un servidor de nombres y ofrecer los servicios que configuremos.

Tras la instalación, la herramienta nos pedirá unos datos de configuración de *ldap-auth-config*:

1. **URI** del servidor: **ldap://nfsnis1.a.ff.es.eu.org/**
2. **dn**: **dc=a,dc=ff,dc=es,dc=eu,dc=org**
3. **versión**: seleccionaremos 3
4. elegimos **si** para almacenar la contraseña en un archivo separado
5. seleccionamos **no** para no exigir autenticación para consultas
6. establecemos el dominio del administrador **cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org**
7. introducimos contraseña que establecimos al instalar slapd

Ahora, configuraremos el demonio slapd a través del comando *sudo dpkg-reconfigure slapd*. Estableceremos la dirección del servidor, un nombre para la organización, pondremos la contraseña anterior, estableceremos el tipo base la que sale por defecto y las demás opciones las dejamos por defecto.

Seguiremos asegurándonos de que el fichero */etc/ldap.conf* contiene nuestro nombre como host, nuestra base y la uri, como se especifica en esta sección.

Para terminar configuraremos la base estableciendo la estructura de esta. El fichero *base.ldif* contendrá la base de nuestra jerarquía, es decir, los usuario y los grupos.:

Para aplicar los cambios usaremos el siguiente comando **ldapadd -x -D cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org -W -f base.ldif** introduciendo la contraseña del administrador. Ahora mismo tenemos listo el servicio ldap, pero antes vamos a configurar un usuario nuevo, crearemos un nuevo usuario del servicio como se indica aquí.

2.2 Máquina cliente - cliente 1

Para empezar, instalaremos las herramientas *libnss-ldap*, *libpam-ldap*, *ldap-utils*, volveremos a configurar *ldap-auth-config* como al principio de la sección 2.1. Nos aseguramos de tener las siguientes entradas en */etc/ldap.conf*:

```
bind_policy soft
pam_password crypt
uri ldap://nfsnis1.a.ff.es.eu.org/
```

Modificamos el contenido del fichero */etc/ldap/ldap.conf* que usa el cliente para las peticiones ldap:

```
BASE      dc=a,dc=ff,dc=es,dc=eu,dc=org      #base DN por defecto
URI       ldap://nfsnis1.a.ff.es.eu.org      #uri del servidor
SIZELIMIT 0                                  #busquedas ilimitadas
TIMELIMIT 0                                  #sin limite de tiempo
DEREF     never                              #para que no referencias los aliases
```

Seguiremos modificando el fichero */etc/nsswitch.conf*, aquí indicamos que puede obtener información de ldap:

```
passwd:      files ldap
group:       files ldap
shadow:      files ldap
```

Para terminar una configuraciones de pam. En primer lugar el fichero */etc/pam.d/common-session* en el que sustituiremos la línea de **password por la siguiente**:

```
password     [success=1 user_unknown=ignore default=die] pam_ldap.so try_first_pass
```

y por último el fichero de */etc/pam.d/common-session* quedará así:

```
session      required pam_mkhomedir.so skel=/etc/skel/ umask=0022
```

. Con esto indicamos que debe montar el directorio del usuario cuando inicie sesión. Para realizar las comprobaciones podemos logearnos desde la máquina cliente con el usuario que hemo creado y veremos como se crea nuestro dictorio, y si añadimos ficheros, también aparecen en el servidor, ya que tenemos configuraddo NFS.

3 REPLICACIÓN LDAP

3.1 Configuración del servidor - Provider

Comenzaremos configurando el **provider**, que en nuestro caso es el servidor ldap.

Crearemos un fichero para configurar el **objectClass** de la base de datos `provider_sync.ldif` con el contenido indicado en el anexo y añadiremos la entrada con el siguiente comando: **sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f provider_sync.ldif**. Pero antes crearemos el directorio de acceslog con **sudo -u openldap mkdir /var/lib/ldap/acceslog**. Este directorio se encarga de mantener un registro de los accesos de una base de datos a otra. Respecto al comando de `ldapadd` usamos el mecanismo externo de SASL con la opción `-Y`.

3.2 Configuración del cliente - Consumer

Creemos el fichero `consumer_sync.ldif` para indicar quien es el proveedor con el contenido que indica el anexo, y lo añadimos con el comando **sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f consumer_sync.ldif**

Para comprobar el correcto funcionamiento ejecutamos **ldapsearch -z1 -LLLQY EXTERNAL -H ldapi:/// -s base -b dc=a,dc=ff,dc=es,dc=eu,dc=org contextCSN** tanto en provider como en consumer y nos aseguramos de obtener un `contextCSN` (indicador de estado de las bases de datos) igual en ambos.

4 TLS

4.1 Provider

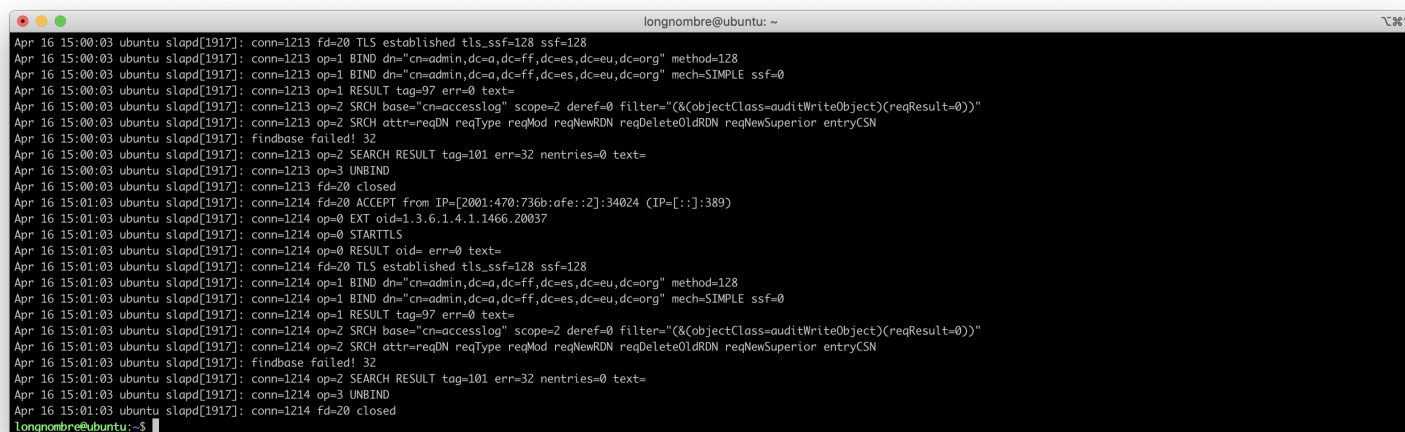
Crearemos unas claves y un certificado de autoridad autofirmado para poder llevar a cabo una conexión TLS segura en LDAP.

Por ahora tenemos una replicación entre provider y consumer, y hemos configurado TLS para autenticación en el provider. Ahora necesitaremos aplicar TLS a la replicación para hacer que esta sea mas seguro. Para ello, crearemos un directorio donde almacenaremos el certificado del consumidor y el nuestro y transferiremos este al consumidor. Para terminar con el master, añadiremos al fichero `/etc/ldap/ldap.conf` la siguiente linea:

```
TLS_REQCERT allow
```

Con esto habilitaremos las solicitudes de certificados para el TLS.

Una última comprobación que podemos realizar es comprobar el fichero `/var/log/syslog` y tras ejecutar un `ldapsearch` ver si la conexión se ha establecido y en tal caso, que se ha haya llevado correctamente.



```
longnombre@ubuntu: ~  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 fd=20 TLS established tls_ssf=128 ssf=128  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 op=1 BIND dn="cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org" method=128  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 op=1 BIND dn="cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org" mech=SIMPLE ssf=0  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 op=1 RESULT tag=97 err=0 text=  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 op=2 SRCH base="cn=accesslog" scope=2 deref=0 filter="(objectClass=auditWriteObject)(reqResult=0)"  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 op=2 SRCH attr=reqDN reqType reqMod reqNewRDN reqDeleteOldRDN reqNewSuperior entryCSN  
Apr 16 15:00:03 ubuntu slapd[1917]: findbase failed! 32  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 op=2 SEARCH RESULT tag=101 err=32 nentries=0 text=  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 op=3 UNBIND  
Apr 16 15:00:03 ubuntu slapd[1917]: conn=1213 fd=20 closed  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 fd=20 ACCEPT from IP=[2001:470:736b:afe::2]:34024 (IP=[::]:389)  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=0 EXT oid=1.3.6.1.4.1.1466.20037  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=0 STARTTLS  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=0 RESULT oid= err=0 text=  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 fd=20 TLS established tls_ssf=128 ssf=128  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=1 BIND dn="cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org" method=128  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=1 BIND dn="cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org" mech=SIMPLE ssf=0  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=1 RESULT tag=97 err=0 text=  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=2 SRCH base="cn=accesslog" scope=2 deref=0 filter="(objectClass=auditWriteObject)(reqResult=0)"  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=2 SRCH attr=reqDN reqType reqMod reqNewRDN reqDeleteOldRDN reqNewSuperior entryCSN  
Apr 16 15:01:03 ubuntu slapd[1917]: findbase failed! 32  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=2 SEARCH RESULT tag=101 err=32 nentries=0 text=  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 op=3 UNBIND  
Apr 16 15:01:03 ubuntu slapd[1917]: conn=1214 fd=20 closed  
longnombre@ubuntu:~$
```

Figure 2: Depliegue del sistema

4.2 Consumer

Lo que haremos es ubicar correctamente los certificados y claves que nos ha proporcionado el proveedor con sus correspondientes permisos y modificaremos unos atributos de `olcDatabase` añadiendo cierta información sobre TLS. Añadiremos la entrada con **sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f consumer_sync_tls.ldif** y reiniciaremos el servicio `slapd`.

Para realizar comprobaciones podemos ejecutar **ldapsearch -x -b "uid=hayk,ou=people,dc=a,dc=ff,dc=es,dc=eu,dc=org" -H ldapi://nfsnis1.a.ff.es.eu.org -ZZ**. Con `-ZZ` obligamos a usar TLS, si obtenemos respuesta, entonces tenemos una conexión segura con TLS.

5 ANEXO 1 - CONFIGURACIÓN

5.1 Fichero interfaces

```
auto lo
iface lo inet6 loopback

auto ens3
iface ens3 inet6 manual

auto ens3.1099
iface ens3.1099 inet6 static
    address 2001:470:736b:aff::5
    netmask 64
    gateway 2001:470:736b:aff::1
    autoconf 0
    vlan-raw-device ens3
```

5.2 Servidor NFS

Tras instalar *nfs-kernel-server* y *nfs-common*, creamos el directorio a exportar, en este caso **/srv/nfs4/home**. A continuación modificamos el fichero **/etc/idmap.d** y añadimos nuestro dominio del servidor NFS, es decir, añadimos "*Domain = a.ff.es.eu.org*".

Continuamos creando el directorio **/home/ldapUsers**, donde ubicaremos los directorios home de cada usuario ldap. Este directorio será el que exportaremos. Para realizar la exportación creamos el directorio **/srv/nfs4/home** y realizamos un **bind** de estos dos directorios para que estén sincronizados. Para que el bind sea permanente lo introduciremos en el fichero **/etc/fstab** de la siguiente forma:

```
/home/ldapUsers /srv/nfs4/home none bind 0 0
```

Para terminar modificamos el fichero de **/etc/exports** con el directorio que pondremos a disposición de los clientes.

```
/srv/nfs4/home cliente1.a.ff.es.eu.org(rw,sync,no_subtree_check,no_root_squash)
```

- **rw:** permitimos tanto escritura como lectura.
- **sync:** al introducir este flag impedimos peticiones cuando hay cambios pendientes.
- **no_subtree_check:** añadimos velocidad negando a NFS que compruebe permisos de directorios padre.
- **no_root_squash:** mantenemos permisos de administrador de los usuarios que lo tenga.

Reiniciamos el demonio con *service nfs-kernel-server restart*.

5.3 Cliente NFS

Tras instalar las herramientas, realizamos un montaje manual con **mount** para comprobar el correcto funcionamiento. En caso positivo, incluimos en **/etc/fstab** la entrada siguiente

```
nfsnis1.a.ff.es.eu.org:/srv/nfs4/home /home nfs4 auto 0 0
```

Indicamos que nos monte el directorio que nos exporta el servidor en **/home**. La opción **auto** lo montará al arranque de la máquina.

5.4 LDAP SERVER

5.4.1 Fichero ldap.conf

```
host nfsnis1.a.ff.es.eu.org
base dc=a,d=ff,dc=es,dc=eu,dc=org
uri ldapi://nfsni1.a.ff.es.eu.org/
rootbinddn cn=admin,dc=a,d=ff,dc=es,dc=eu,dc=org
ldap_version 3
bin_policy soft
```

5.4.2 Fichero base.ldif

```
dn: ou=people ,dc=a ,d=ff ,dc=es ,dc=eu ,dc=org
objectClass: organizationalUnit
ou: people
```

```
dn: ou=groups ,dc=a ,d=ff ,dc=es ,dc=eu ,dc=org
objectClass: organizationalUnit
ou: groups
```

5.4.3 New User

Crearemos el fichero hayk.ldif con el siguiente contenido:

```
dn: uid=hayk ,ou=people ,dc=a ,dc=ff ,dc=es ,dc=eu ,dc=org
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: hayk
cn: hayk
sn: kocharyan
userPassword: haykk99
displayName: hayk
loginShell: /bin/bash
uidNumber: 2010
gidNumber: 2010
homeDirectory: /home/ldapUsers/hayk

dn: cn=grupoDeHayk ,ou=groups ,dc=a ,dc=ff ,dc=es ,dc=eu ,dc=org
objectClass: posixGroup
cn: grupoDeHayk
gidNumber: 2010
```

Para aplicar los comandos usaremos **ldapadd -x -D cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org -W -f hayk.ldif**.

5.5 Replicación

5.5.1 Provider sync

```
# Add indexes to the frontend db.
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryCSN eq
-
add: olcDbIndex
olcDbIndex: entryUUID eq

#Load the syncprov and accesslog modules.
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
-
add: olcModuleLoad
olcModuleLoad: accesslog

# Accesslog database definitions
dn: olcDatabase={2}mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: {2}mdb
olcDbDirectory: /var/lib/ldap/accesslog
olcSuffix: cn=accesslog
olcRootDN: cn=admin,dc=example,dc=com
olcDbIndex: default eq
olcDbIndex: entryCSN,objectClass,reqEnd,reqResult,reqStart

# Accesslog db syncprov.
dn: olcOverlay=syncprov,olcDatabase={2}mdb,cn=config
```

```

changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE
olcSpReloadHint: TRUE

# syncrepl Provider for primary db
dn: olcOverlay=syncprov,olcDatabase={1}mdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE

# accesslog overlay definitions for primary db
dn: olcOverlay=accesslog,olcDatabase={1}mdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcAccessLogConfig
olcOverlay: accesslog
olcAccessLogDB: cn=accesslog
olcAccessLogOps: writes
olcAccessLogSuccess: TRUE
# scan the accesslog DB every day, and purge entries older than 7 days
olcAccessLogPurge: 07+00:00 01+00:00

```

5.5.2 Consumer sync

```

dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID eq
—
add: olcSyncRepl
olcSyncRepl: rid=0
provider=ldap://nfsnis1.a.ff.es.eu.org
bindmethod=simple
binddn="cn=admin,dc=a,dc=ff,dc=es,dc=eu,dc=org"
credentials=longnombre
searchbase="dc=a,dc=ff,dc=es,dc=eu,dc=org"
logbase="cn=accesslog"
logfilter="(&(objectClass=auditWriteObject)(reqResult=0))"
schemachecking=on
type=refreshAndPersist
retry="60_+"
syncdata=accesslog
—
add: olcUpdateRef
olcUpdateRef: ldap://nfsnis1.a.ff.es.eu.org

```

Nos aseguraremos de rellenar bien los campos del atributo **olcSuncRepl** ya que sin esto no funciona la replicación.

5.6 Certificado de autoridad y claves

```

// instalamos herramientas
sudo apt install gnutls-bin ssl-cert
// Generamos una clave privada para el CA
sudo sh -c "certtool --generate-privkey > /etc/ssl/private/cakey.pem"
// Creamos un fichero para definir el CA
sudo nano /etc/ssl/ca.info
// su contenido:
cn = nfsnis1

```

```

ca
cert_signing_key
// creamos el CA autofirmado
sudo certtool --generate-self-signed \
--load-privkey /etc/ssl/private/cakey.pem \
--template /etc/ssl/ca.info \
--outfile /etc/ssl/certs/cacert.pem
// Creamos una clave privada para el servidor
sudo certtool --generate-privkey --bits 1024 --outfile /etc/ssl/private/server_slapd_key.pem
// creamos el fichero de información sobre el CA (validez de 10 años) e introducimos su contenido
sudo nano /etc/ssl/server.info
    organization = nfsnis1
    cn = nfsnis1.a.ff.es.eu.org
    tls_www_server
    encryption_key
    signing_key
    expiration_days = 3650
// Creamos el certificado del servidor
sudo certtool --generate-certificate \
--load-privkey /etc/ssl/private/server_slapd_key.pem \
--load-ca-certificate /etc/ssl/certs/cacert.pem \
--load-ca-privkey /etc/ssl/private/cakey.pem \
--template /etc/ssl/server.info \
--outfile /etc/ssl/certs/server_slapd_cert.pem
// Cambiamos permisos
sudo chgrp openldap /etc/ssl/private/server_slapd_key.pem
sudo chmod 0640 /etc/ssl/private/server_slapd_key.pem
sudo gpasswd -a openldap ssl-cert
// Y por último, reiniciamos el demonio
sudo systemctl restart slapd.service

```

Para añadir los certificados al servicio ldap creamos un fichero ldif con el siguiente contenido:

```

dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/server_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/server_slapd_key.pem

```

Y finalmente, con ldapmodify le decimos a slapd sobre el uso de TLS en nuestra base. **sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif**

5.7 Replicación y TLS

5.7.1 Provider

```

// creamos el directorio del consumidor, en este caso la maquina cliente1
sudo mkdir cliente1-ssl
cd cliente1-ssl
// creamos clave privada del consumidor
sudo certtool --generate-privkey \
--bits 1024 \
--outfile cliente1_slapd_key.pem
// El fichero .info para el CA y metemos su contenido.
sudo nano cliente1.info
    organization = nfsnis1
    cn = cliente1.a.ff.es.eu.org
    tls_www_server
    encryption_key
    signing_key
    expiration_days = 3650
// Creamos el certificado del consumidor
sudo certtool --generate-certificate \

```

```

—load-privkey cliente1_slapd_key.pem \
—load-ca-certificate /etc/ssl/certs/cacert.pem \
—load-ca-privkey /etc/ssl/private/cakey.pem \
—template cliente1.info \
—outfile cliente1_slapd_cert.pem
// copiamos nuestro certificado
sudo cp /etc/ssl/certs/cacert.pem .
// enviamos el contenido al cliente1
scp -r cliente1-ssl longnombre@cliente1.a. ff. es. eu. org:/home/longnombre

```

5.7.2 Consumer

```

// instalamos herramientas
sudo apt install ssl-cert
// ubicamos los ficheros correctamente
sudo gpasswd -a openldap ssl-cert
sudo cp cliente1_slapd_cert.pem cacert.pem /etc/ssl/certs
sudo cp cliente1_slapd_key.pem /etc/ssl/private
sudo chgrp openldap /etc/ssl/private/cliente1_slapd_key.pem
sudo chmod 0640 /etc/ssl/private/cliente1_slapd_key.pem
sudo systemctl restart slapd.service

```

Ahora modificaremos la base de datos con la siguiente configuración:

```

dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
—
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/cliente1_slapd_cert.pem
—
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/cliente1_slapd_key.pem

```

Aplicamos los cambios: sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f /etc/ssl/certinfo.ldif

6 ANEXO 2 - SCRIPTS USADOS

6.1 ENCENDIDO DE MÁQUINAS

```
#!/bin/bash
```

```

echo " ... Procediendo al montaje de las maquinas"
virsh -c qemu:///system define /misc/alumnos/as2/as22019/a757715/oAFF2.xml
virsh -c qemu:///system define /misc/alumnos/as2/as22019/a757715/oAFF3.xml
virsh -c qemu:///system define /misc/alumnos/as2/as22019/a757715/oAFF4.xml
virsh -c qemu:///system define /misc/alumnos/as2/as22019/a757715/oAFF5.xml
virsh -c qemu:///system define /misc/alumnos/as2/as22019/a757715/oAFF6.xml
virsh -c qemu:///system define /misc/alumnos/as2/as22019/a757715/orouterA.xml

echo " ... Procediendo al encendido"
virsh -c qemu:///system start oAFF2
virsh -c qemu:///system start oAFF3
virsh -c qemu:///system start oAFF4
virsh -c qemu:///system start oAFF5
virsh -c qemu:///system start oAFF6
virsh -c qemu:///system start routerA

```

```
echo " ... Listando ... "
```

6.2 APAGADO DE MAQUINAS

```
#!/bin/bash
```

```

echo " ... procediendo al apagado"
host=2001:470:736b:aff::

for i in {4..1}; do

```



```
machine=$host$i
echo $machine
ssh $machine 'doas _shutdown _h_now'
sleep 2
echo "...maquina_$i_apagada"
done
```

```
echo "...apagando_maquinas_ubuntu"
virsh -c qemu:///system shutdown oAFF5
virsh -c qemu:///system shutdown oAFF6
echo "..apagadas"
sleep 2
```

```
echo "...destruyendo_maquinas"
```

```
virsh -c qemu:///system destroy oAFF2
virsh -c qemu:///system destroy oAFF3
virsh -c qemu:///system destroy oAFF4
virsh -c qemu:///system destroy oAFF5
virsh -c qemu:///system destroy oAFF6
virsh -c qemu:///system destroy orouterA
```

```
echo "...quitando_maquinas"
```

```
virsh -c qemu:///system undefine oAFF2
virsh -c qemu:///system undefine oAFF3
virsh -c qemu:///system undefine oAFF4
virsh -c qemu:///system undefine oAFF5
virsh -c qemu:///system undefine oAFF6
virsh -c qemu:///system undefine orouterA
```