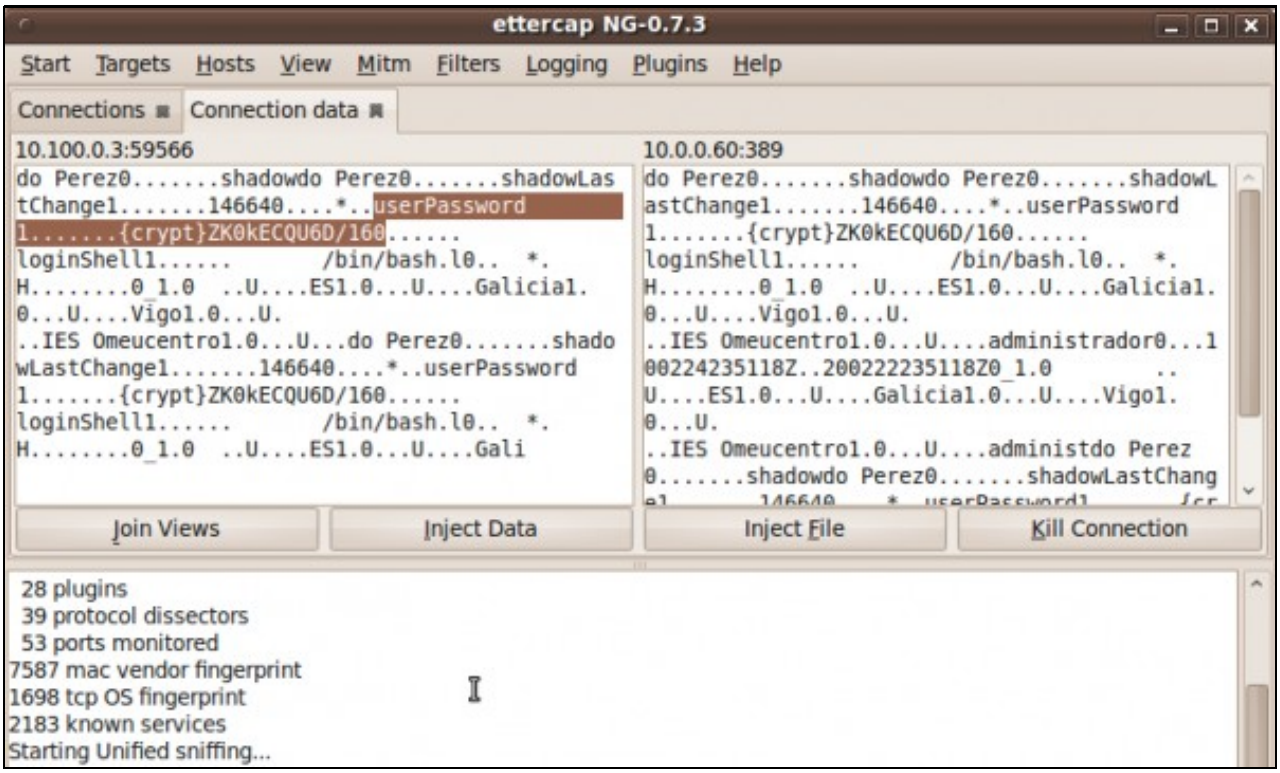
SSL

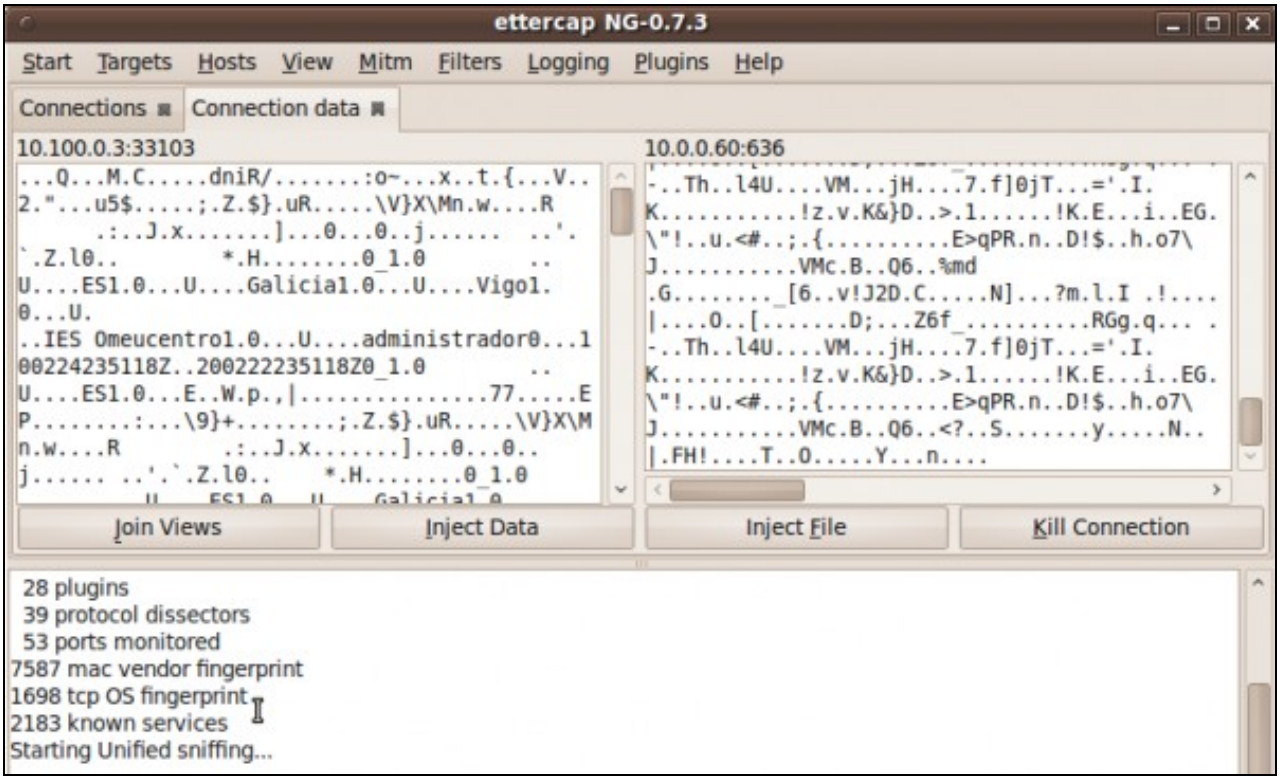
Sumario

¿Por que usar TLS/SSL?

Cando usamos un servidor LDAP para autenticar os usuarios dun dominio, é conveniente que a comunicación entre o cliente e o servidor no proceso de autenticación se faga de forma segura. A razón é simple: se o tráfico de autenticación faise en claro, calquera pode capturar os paquetes intercambiados entre cliente e servidor para obter o contrasinal do usuario. O método utilizado para a codificación do contrasinal admite varias opcións, e no noso caso úsase por defecto o algoritmo **CRYPT**, que ten un nivel de seguridade bastante aceptable pero sempre é susceptible a ataques usando dicionarios de contrasinais se os contrasinais dos usuarios non son suficientemente fortes, polo que sería conveniente establecer unha seguridade maior para o intercambio desta información. Na seguinte imaxe móstrase a captura usando o programa ettercap dos paquetes intercambiados entre un cliente e un servidor LDAP nunha autenticación non segura:



Podemos usar **TLS**/**SSL** (*Transport Layer Secutiry/Secure Sockets Layer*) para cifrar a sesión entre cliente e servidor, de forma que será máis difícil (nunca imposible, por suposto) capturar a información que se intercambian no proceso de autenticación e, sobre todo, o contrasinal do usuario. Na seguinte imaxe móstrase a captura dos paquetes intercambiados entre un cliente e un servidor LDAP nunha autenticación segura con *TLS/SSL*:



Requirimentos para o uso de TLS/SSL

Para configurar o noso servidor LDAP para usar TLS/SSL no proceso de autenticación, precisamos instalar e configurar unha serie de compoñentes:

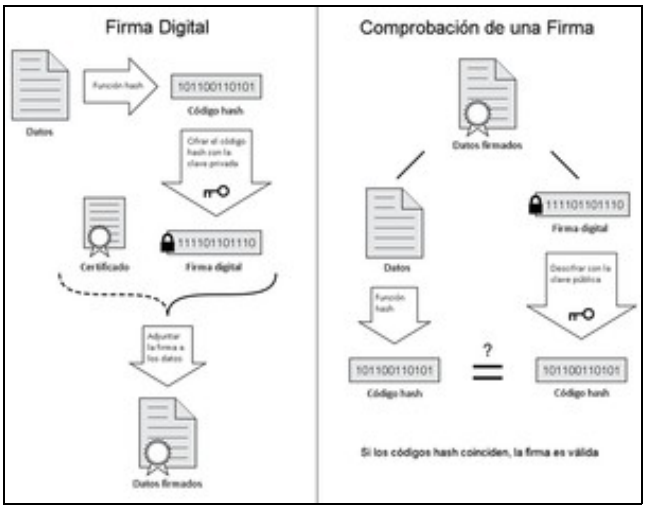
- **Servidor de DNS**: Para poder cifrar a comunicación entre o cliente e o servidor, teremos que xerar un certificado dixital para o servidor asociado a un nome completo de dominio (**FQDN**) que asignaremos ao servidor. Este nome de DNS será utilizado polos clientes para conectarse ao servidor LDAP. Polo tanto, o primeiro que teremos que facer é instalar un servidor de DNS no que ese nome completo estará asociado á dirección IP do servidor LDAP. Dirixirémonos aos seguintes apartados:
  - Introdución ao servizo DNS.
  - Instalación e configuración do servizo DNS con Ubuntu Server.

- **Autoridade de certificación**: A continuación, teremos que crear unha autoridade de certificación para crear un certificado dixital para o servidor no que os clientes terán que confiar. Nos seguintes apartados indícase os pasos que teremos que seguir.

Introdución aos certificados dixitais

Sen entrar en moitos detalles, imos facer un breve resumo dos conceptos básicos dos certificados dixitais para poder comprender os pasos que levaremos a cabo nos seguintes apartados.

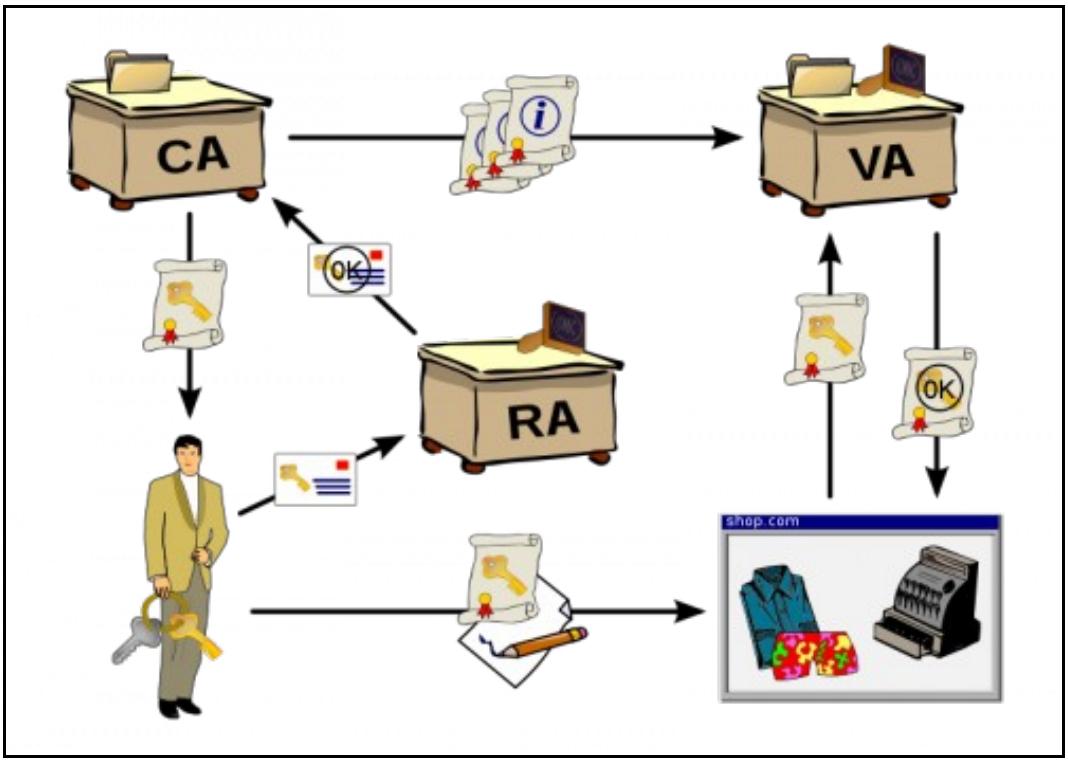
Un certificado dixital é un documento dixital (unha ristra de bytes) mediante a que unha entidade fiable, coñecida como autoridade de certificación (aínda que nos refiriremos a ela habitualmente como *CA*), garante que unha chave pública correspóndese con unha entidade concreta. Con entidade concreta moitas veces nos referimos a un nome de equipo ou un dominio de DNS, e desta forma poderemos asegurarnos de que nos estamos conectando a o equipo auténtico e que a información que enviamos só poderá ser recibida por ese equipo. O formato estándar que máis se usa para os certificados dixitais é o X.509, que é o que usaremos no noso caso. Segundo este formato, o certificado con unha serie de campos entre os que destacan a versión, o número de serie, a validez do certificado, o seu emisor (a *CA* que o emite), o suxeito para o que se emite o certificado e a chave pública do suxeito.



Proceso de xeración e comprobación dunha sinatura dixital

Os certificados dixitais son utilizados nos métodos de cifrado asimétricos ou de chave pública, que baséanse na utilización dun par de chaves: A **chave pública**, que como o seu nome indica é pública e pode ser coñecida por calquera, e a **chave privada** que só pode ser coñecida polo seu propietario. Estas chaves teñen as propiedades de que a información cifrada usando a chave pública só pode ser descifrada coa chave privada, mentres que unha información cifrada coa chave privada só pode ser descifrada usando a chave pública. Desta forma, cando un equipo cifra unha información utilizando a chave pública do destinatario (que obterá dun certificado dixital), só o destinatario poderá descifrar a mensaxe coa súa chave privada (que só el coñece), e polo tanto estamos garantindo a *confidencialidade* da información. Por outra banda, cando un equipo cifra unha mensaxe coa súa chave privada, calquera pode descifralo usando a súa chave pública (polo que non garantimos así a confidencialidade), pero estamos garantindo a *identificación* e *autenticación* do remitente (xa que se podemos descifralo coa chave pública quere dicir que o remitente coñece a chave privada), dando lugar á sinatura dixital.

O uso de certificados dixitais nos dous equipos que establecen unha comunicación, e o uso dos métodos de cifrado de chave pública, permiten garantir todos os requirimentos dunha conexión segura. A combinación dos certificados dixitais e as entidades necesarias para a súa emisión cos métodos de cifrado e chave pública xunto co hardware e as políticas de seguridade que permiten levar a cabo as operacións de cifrado de xeito seguro conforman o que se coñece como a Infraestrutura de Chave Pública (PKI). Na seguinte imaxe móstranse os compoñentes básicos dunha PKI:



Un usuario solicita un certificado dixital a unha autoridade de rexistro (*RA*), que se encarga da verificar a autenticidade do usuario, e enviar a súa verificación á autoridade de certificación (*CA*), que emite o certificado para o usuario. Con este certificado, o usuario pode firmar dixitalmente documentos, xa que cifrándoos coa súa chave privada e enviando o seu certificado a autoridade de validación (*VA*) poderá confirmar que realmente é o usuario o que emitiu o documento.

Creación dos certificados dixitais

Unha vez aclarados os conceptos básicos sobre os certificados dixitais, veremos que é o que imos facer no noso caso. O método de cifrado TLS/SSL utiliza un método de cifrado de chave pública para a autenticación do servidor (e tamén se podería facer do cliente, aínda que nós non o faremos) para xerar e intercambiar a partir de aí unha chave privada compartida e usar un método de cifrado *simétrico ou de chave privada* (no que se cifra e descifra a información coa mesma chave privada que só o emisor e receptor coñecen). Os pasos que seguiremos son os seguintes:

- Crearemos unha autoridade de certificación (CA).
- Crearemos unha solicitude de firma de certificado (CSR) para que a CA cree o certificado para o servidor (asociado ao nome DNS do servidor).
- Xeraremos coa CA o certificado do servidor a partir da CSR.
- Teremos que copiar no equipo cliente o certificado da CA, para que cando reciba o certificado do servidor confíe nel ao estar emitido por esa CA.

Crear a Autoridade de Certificación (CA)

Primeiro, creamos os directorios para almacenar os certificados da CA e os ficheiros relacionados:

```
sudo mkdir /etc/ssl/CA
sudo mkdir /etc/ssl/newcerts
```

Creamos dous ficheiros que a CA precisará para manter un número de serie que lle asignará a cada certificado e almacenar os certificados emitidos:

```
sudo sh -c "echo '01' > /etc/ssl/CA/serial"
sudo touch /etc/ssl/CA/index.txt
```

No ficheiro de configuración da CA /**etc/ssl/openssl.cnf**, modificaremos os seguintes parámetros dentro da sección *[CA_default]*:

```
dir             = /etc/ssl/             # Where everything is kept
database        = $dir/CA/index.txt     # database index file.
certificate     = $dir/certs/cacert.pem # The CA certificate
serial          = $dir/CA/serial        # The current serial number
```

Creamos o certificado raíz para a propia CA, que será firmado por si mesma:

```
openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days 3650
```

teremos que introducir un contrasinal para a CA (podemos poñer *abc123.*), e os datos do certificado. A continuación móstrase un exemplo para estes datos. É importante ter en conta que o que poñamos en *Organization Name*, deberá ser o mesmo valor que loo poñamos neste mesmo campo no certificado do servidor:

```
Generating a 1024 bit RSA private key
......++++++
........++++++
unable to write 'random state'
writing new private key to 'cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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]:ES
State or Province Name (full name) [Some-State]:Galicia
Locality Name (eg, city) []:
```

```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES calquera
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:server00.iescalquera.local
Email Address []:
```

e instalamos nos directorios da CA tanto a chave privada como o certificado creado:

```
sudo mv cakey.pem /etc/ssl/private/
sudo mv cacert.pem /etc/ssl/certs/
```

Xenerar a solicitude de firma do certificado (*CSR*)

En primeiro lugar teremos que crear unha chave para xerar a CSR, que será almacenada no ficheiro **server.key**. Teremos que introducir un contrasinal que será necesario para abrir esta chave (Como exemplo, podemos poñer o mesmo contrasinal *abc123.*):

```
openssl genrsa -des3 -out server.key 1024
```

O problema que temos con esta chave que acabamos de crear é que para poder abrila fai falta proporcionar o contrasinal que lle asignamos, e entón cada vez que se arrancara o servidor LDAP habería que introducir este contrasinal para que puidese ter acceso á chave privada do servidor, e isto supón un problema xa que calquera reinicio do servizo obriga a unha intervención manual. Por iso, o que imos facer é crear a partir da chave xa creada unha chave que non requira contrasinal:

```
openssl rsa -in server.key -out server.key.insecure
```

E gardamos en *server.key* a chave sen contrasinal, que será a que usaremos:

```
mv server.key server.key.secure
mv server.key.insecure server.key
```

E por último creamos o CSR:

```
openssl req -new -key server.key -out server.csr
```

Introduciremos os datos necesarios para a solicitude do certificado, destacando o *Organization Name*, que deberá coincidir co que introducimos para a CA, e o *Common Name*, que deberá ser o nome DNS do servidor para o que emitiremos o certificado:

```
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Galicia
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES calquera
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:server00.iescalquera.local
Email Address []:

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

A CSR será almacenada no ficheiro **server.csr**, que xa pode ser enviada á autoridade de certificación para que xenere o certificado.

Xerar o certificado a partir do CSR

Ah!! pero se a autoridade de certificación tamén somos nós!! Ben, pois imos crear un certificado a partir da CSR:

```
sudo openssl ca -in server.csr -config /etc/ssl/openssl.cnf
```

Primeiro pedirásenos o contrasinal da CA (o que asignamos cando creamos o certificado da CA, no noso caso *abc123.*), e a continuación mostrásenos os datos do certificado que se vai xerar (tomados da CSR):

```
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Mar  4 23:25:23 2010 GMT
            Not After : Mar  4 23:25:23 2011 GMT
```

```
        Subject:
            countryName               = ES
            stateOrProvinceName       = Galicia
            organizationName          = IES calquera
            commonName                = server00.iescalquera.local
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                DF:FC:73:0D:36:B0:AF:DA:47:F7:E3:57:F9:41:FD:FF:88:AF:17:AE
            X509v3 Authority Key Identifier:
                keyid:01:C8:B2:AD:1B:B7:86:45:3E:CA:37:CC:C1:95:8E:A8:22:C3:D1:9B

Certificate is to be certified until Mar  4 23:25:23 2011 GMT (365 days)
```

E procedemos a asinar... (respondemos que si (y) ás dúas preguntas de confirmación).

Listo!! Copiamos todo o texto entre as liñas -----BEGIN CERTIFICATE----- and ----END CERTIFICATE----- (incluíndo estas dúas liñas), e o pegamos no ficheiro *server.crt*. Por exemplo, este ficheiro pode conter algo así:

```
-----BEGIN CERTIFICATE-----
MIICpzCCAhCgAwIBAgIBATANBgkqhkiG9w0BAQUFADBbMQswCQYDVQQGEwJFUzEQ
MA4GA1UECBMHR2FsaWNpYTEVMBMGA1UEChMMSUVTIGNhbHF1ZXJhMSMwIQYDVQQD
ExpzZXJ2ZXIwMC5pZXNjYWxxdWVyYS5sb2NhbDAeFw0xMDAzMDQyMzI1MjNaFw0x
MTAzMDQyMzI1MjNaMFsxCzAJBgNVBAYTAkVTMRAwDgYDVQQIEwdHYWxpY2lhMRUw
EwYDVQQKEwxJRVMgY2FscXVlcmExIzAhBgNVBAMTGnNlcnZlcjAwLmllc2NhbHF1
ZXJhLmxvY2FsMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/QWqoi12reBRA
/3p6+KyWTAoN3XqLU8VaNhpAAP4LTRuuzeeCKxkPyj2QZk+rWehmqkqbwX6Zdrqi
BSfeKuoRokTV7e2bbMJmaomEbvez5bwr7sDSXl2UyFhVyJWtQBkI8m2pkqjWt9Fn
2OotV+c43HNncXN3/mGoVwpE7OMivwIDAQABo3sweTAJBgNVHRMEAjAAMCwGCWCG
SAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4E
FgQU3/xzDTawr9pH9+NX+UH9/4ivF64wHwYDVR0jBBgwFoAUAciyrRu3hkU+yjfM
wZWOqCLD0ZswDQYJKoZIhvcNAQEFBQADgYEAVHDWexRWbz6nPWVA+x/4KaXA9KaE
atZ1cu2Mep+29duZyAFcQEf4pivXCallmkmbAhurpUH61SLFHOb7YHl71EPLvru0
U3kDx48wSDGqBzdCKWhoh1SBrFryxlovEredZ44q/1AxldJ8py9r77e2kqJ7u+TC
6v0/CnJRUYvWZh0=
-----END CERTIFICATE-----
```

Copiamos o certificado e a chave ao directorio de almacenamento da CA:

```
sudo cp server.crt /etc/ssl/certs
sudo cp server.key /etc/ssl/private
```

Configuración do servidor LDAP

Unha vez que temos creados o certificado e chave para o servidor e o certificado da CA, temos que o servidor LDAP para que faga uso deles nas conexións seguras:

```
ldapmodify -x -D cn=admin,cn=config -W
```

Introduciremos como contrasinal *1234*, e pegaremos os seguintes datos:

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/server.crt
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/server.key
```

E prememos as teclas *Control+D* para procesar os datos introducidos.

Editamos o ficheiro */etc/default/slapd* para establecer no parámetro **SLAPD_SERVICES** o seguinte valor:

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

Permitimos o acceso ao certificado ao usuario *openldap*, xa que é o usuario co que se executa o servidor LDAP:

```
sudo addgroup ssl-cert
sudo adduser openldap ssl-cert
sudo chmod 750 /etc/ssl/private
sudo chgrp ssl-cert /etc/ssl/private
sudo chmod 640 /etc/ssl/private/server.key
sudo chgrp ssl-cert /etc/ssl/private/server.key
```

E reiniciamos o servizo **slapd**:

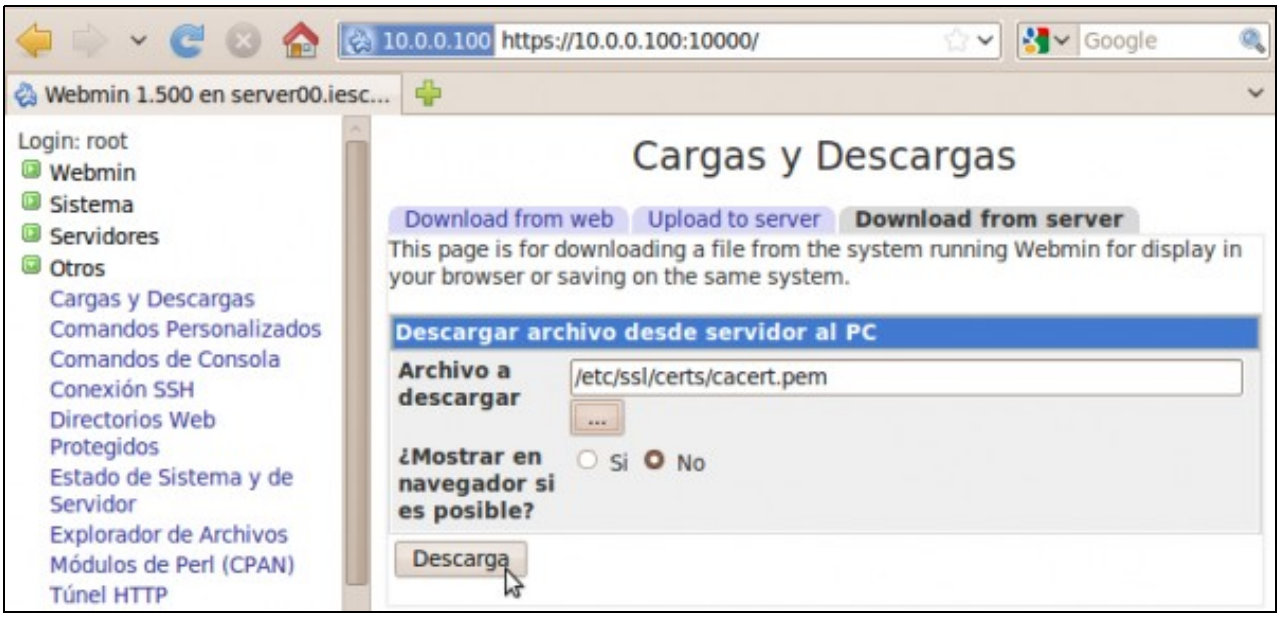
```
sudo /etc/init.d/slapd restart
```

Configuración do cliente LDAP

Agora quédanos configurar o equipo cliente para que realice a autenticación co servidor LDAP de forma segura, usando o protocolo **ldaps** en lugar de **ldap**:

- Temos que copiar o certificado da CA no cliente, para que así este confíe no certificado que lle enviará o servidor, xa que está emitido por esa CA. O problema é que este certificado témolo almacenado no servidor, no ficheiro */etc/ssl/certs/cacert.pem*; ¿como o copiamos ao equipo cliente?:
    - Facendo uso de scp
    - Sacándolle partido ao webmin.

      Conectámonos dende o cliente ao webmin do servidor (*http://10.0.0.100:10000* sería a URL a introducir no navegador dende o equipo cliente no noso caso), e entramos no módulo de **Cargas y Descargas** que atoparemos dentro da categoría de **Otros**. Picamos na pestana de **Download from server**, seleccionamos o ficheiro e picamos en **Descarga**:



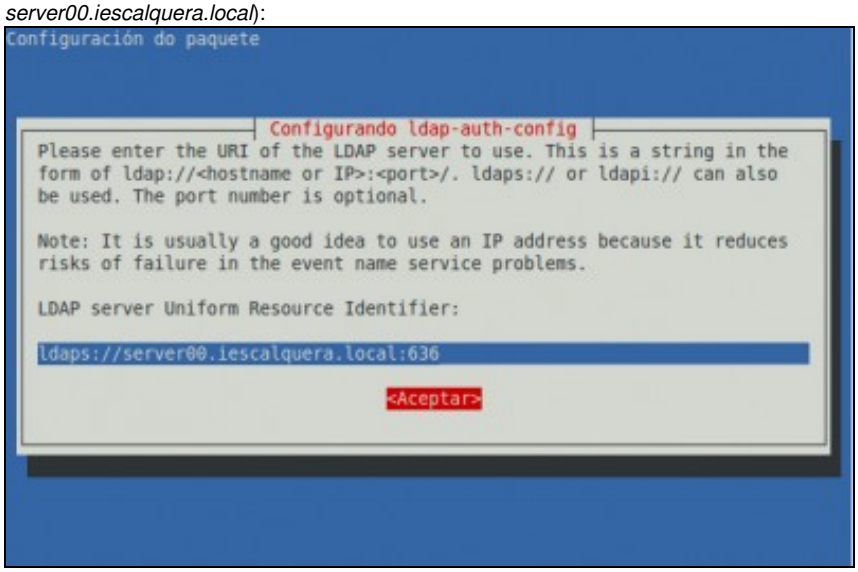
Seguramente Firefox descargará o ficheiro automaticamente na carpeta **Descargas** dentro da carpeta persoal do usuario co que iniciáramos a sesión no equipo cliente. Teremos que movelo de aí á carpeta */etc/ssl*, e podemos facelo introducindo o seguinte comando (supoñendo que nos atopamos situados na carpeta persoal do usuario):

```
sudo mv Descargas/cacert.pem /etc/ssl
```

- Temos que reconfigurar o paquete de autenticación ldap para que se conecte ao servidor por *ldaps*. Executamos o comando:

```
sudo dpkg-reconfigure ldap-auth-config
```

Introduciremos os mesmos datos que xa metéramos na configuración inicial e que xa nos aparecerán por defecto (ver Configuración do cliente LDAP en caso de dúbidas), salvo na **URI** do servidor LDAP, xa que deberemos introducir a que usa o protocolo seguro: **ldaps:**//**NomeDNSServidor:636** (onde **NomeDNSServidor** debe coincidir co nome de DNS para o que se fixo o certificado, no noso caso *server00.iescalquera.local*):



- E xa por último (agora si que rematamos!!), editamos o ficheiro de configuración do cliente LDAP */etc/ldap.conf* para activar os seguintes parámetros (Están cara o final do ficheiro, no apartado *OpenLDAP SSL mechanism*):

```
ssl on
tls_cacertfile /etc/ssl/cacert.pem
```

O cliente xa deberá tomar os usuarios do servidor LDAP usando conexións cifradas usando TLS/SSL.

-- Antonio de Andrés Lema e Carlos Carrión Álvarez