

UOC
Enginyeria Tècnica d'Informàtica de Sistemes

TFC: Autoritat de certificació PKI amb serveis en línia
MEMÒRIA

Eugeni Queralt Monné, juny 2011

Índex

- 1 [Definició i descripció del TFC](#)
- 2 [Objectiu.](#)
- 3 [Preparació de l'entorn](#)
 - 3.1 [Maquinari i sistema operatiu](#)
 - 3.2 [Instal·lació Openssh](#)
 - 3.3 [Instal·lació d'Apache2](#)
 - 3.4 [Instal·lació MySQL5](#)
 - 3.5 [Instal·lació PHP5](#)
 - 3.6 [Instal·lació phpMyAdmin](#)
- 4 [Creació de la CA](#)
 - 4.1 [Generar una petició de signatura de certificat \(CSR\)](#)
 - 4.2 [Generació d'un certificat auto-signat](#)
 - 4.3 [Instal·lació del certificat](#)
 - 4.4 [Creació i configuració de la nostra CA](#)
- 5 [Operacions de la CA](#)
 - 5.1 [Generar peticions de certificat](#)
 - 5.2 [Signar peticions de certificat](#)
 - 5.3 [Revocar certificats](#)
 - 5.4 [Generar la llista de certificats revocats](#)
 - 5.5 [Mostrar la llista de revocats](#)
 - 5.6 [Mostrar si un certificat està revocat](#)
 - 5.7 [Renovar un certificat](#)
 - 5.8 [Mostrar propietari](#)
- 6 [Configuració de l' HTTPS](#)
- 7 [Permisos per l'usuari web \(www-data\)](#)
- 8 [Aplicació web](#)
 - 8.1 [Inici i registre de clients](#)
 - 8.2 [Peticions de signatura](#)
 - 8.3 [Signatura de peticions](#)
 - 8.4 [Emissió de certificats PKCS#12](#)
 - 8.5 [Consulta de la llista de revocació](#)
 - 8.6 [Revocació de certificats](#)
 - 8.7 [Generació de la llista de revocació](#)
 - 8.8 [Introducció manual de la petició en format PKCS#10](#)
- 9 [Autenticació de les funcions administratives a través del certificat digital expedit per la nostra CA](#)
- 10 [Autenticació dels clients amb certificat](#)
- 11 [Estructura de la Base de Dades](#)
- 12 [Implementació i estructura del programari](#)
- 13 [Bibliografia](#)
- 14 [Webgrafia](#)

Definició i descripció del TFC



Crear una autoritat de certificació amb serveis en línia, amb interfície web amb funcions d'administració i funcions de certificació. En principi havia considerat la recomanació d'utilitzar java i les llibreries associades, però plantejar-me la situació d'optar per un altre llenguatge que pugui millorar les possibilitats de la interfície web, com el PHP.

Investigant (el poc temps que he pogut) quines llibreries i eines del camp seguretat hi havia, de cara a fer el servidor més segur i implementar els diferents requeriments de la CA. La meva conclusió, va ser que valia la pena provar-ho amb aquest llenguatge, principalment per diversos motius:

- Semblava que aquest llenguatge en les seves últimes versions s'integra molt bé (igual que el java) amb el servidor MySQL i el servidor Apache, és interpretat per tots els navegadors utilitzats habitualment i incorpora una bona bateria de funcions de seguretat. A més, permet treballar en forma estructurada i a la vegada disposa de mecanismes per l'orientació a objectes per a projectes grans, tot i que aquest no ho és massa.
- D'altra banda, reconeixia la meua manca experiència en aquest llenguatge i m'interessava aprendre aspectes avançats d'aquest (els continguts del projecte m'ho permetran). Considero que ja he fet prou assignatures en java a la UOC i no he tocat prou contingut per programar en llenguatges interpretats. Per mi és més important aprendre que realitzar un bon projecte, encara que m'esforçaré en aconseguir-ho.

Objectiu




Elaborar una autoritat de certificació CA amb serveis en línia implementada dins un sistema Linux amb MySQL com a sistema de base de dades escollida, una interfície programada amb llenguatge PHP i un servidor Apache per a servir la interfície esmentada. Implementar l'aplicació partint de protocols segurs (SSL) sempre que sigui possible.

- Sistema : Linux celeron 2.6.35-22-generic-pae #35-Ubuntu SMP Sat Oct 16 22:16:51 UTC 2010 i686
- Servidor web: Apache2 (Apache/2.2.16 (Ubuntu))
- Protocols de seguretat: OpenSSL (OpenSSL 0.9.8o 01 Jun 2010)
- Llenguatge de programació: PHP5 (5.3.3-1ubuntu9.3)
- Servidor de base de dades: MySQL (MySQL:5.1.49)

Respecte al sistema (celeron) en principi s'hi va instal·lar tot el programari esmentat, malgrat això, per implementar la interfície web i els diferents mòduls s'ha hagut de migrar a un maquinari més potent, tot i així, manté les funcionalitats i la portabilitat desitjades.

Preparació de l'entorn



Maquinari i sistema operatiu

- Sistema : Linux celeron 2.6.35-22-generic-pae #35-Ubuntu SMP Sat Oct 16 22:16:51 UTC 2010 i686
- Servidor web: Apache2 (Apache/2.2.16 (Ubuntu))
- Protocols de seguretat: OpenSSL (OpenSSL 0.9.8o 01 Jun 2010)
- Llenguatge de programació: PHP5 (5.3.3-1ubuntu9.3)
- Servidor de base de dades: MySQL (MySQL:5.1.49)

Respecte al sistema (celeron) en principi s'hi va instal·lar tot el programari esmentat, malgrat això, per implementar la interfície web i els diferents mòduls s'ha hagut de migrar a un maquinari més potent, tot i així, manté les funcionalitats i la portabilitat desitjades.

Partim d'una instal·lació prèvia del sistema operatiu servidor Ubuntu Server 10.10

Instal·lació bàsica (sense xifrar)

Instal·lació:

- La instal·lació pot requerir un monitor fins la instal·lació de l'administració remota
- El meu servidor està commutat amb el monitor i teclat del l'ordinador d'escriptori, cosa que facilita molt la gestió. Tot i així el configurarem per treballar de forma remota, que serà la forma desitjada.

Instal·lació Openssh



- Administració remota del servidor: Instal·larem el Openssh per poder gestionar de forma remota el nostre servidor:
 - `sudo apt-get install openssh-server openssh-client`
 - habilitarem aquest servei al port 2222 (evitar atacs automatitzats al port 22):
 - `sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original`
 - `sudo chmod a-w /etc/ssh/sshd_config.original`
 - `sudo nano /etc/ssh/sshd_config`; canviant port 22 (per defecte) a port 2222
 - Connexió remota al servidor:

```
$ ssh usuari@192.168.1.35 -p 2222
```

```
The authenticity of host '[192.168.1.35]:2222 ([192.168.1.35]:2222)' can't be established.
```

```
RSA key fingerprint is 53:99:b2:ae:6f:e8:49:8f:96:ab:4a:09:dd:0f:5d:39.
```

```
Are you sure you want to continue connecting (yes/no)?
```

- Contestarem "yes"

```
Warning: Permanently added '[192.168.1.35]:2222' (RSA) to the list of known hosts.
```

```
keni@athlon:~$ ssh usuari@192.168.1.35 -p 2222
```

```
usuari@192.168.1.35's password:
```

```
Linux celeron 2.6.35-22-generic-pae #33-Ubuntu SMP Sun Sep 19 22:14:14 UTC 2010 i686 GNU/Linux
```

Ubuntu 10.10

Welcome to Ubuntu!

* Documentation: <https://help.ubuntu.com/>

Last login: Mon Mar 14 10:32:10 2011

usuari@celeron:~\$

- A partir d'aquí connectarem de forma remota des de la xarxa local (o des de la xarxa si activem el port forwarding del router)
- Farem la IP estàtica, del servidor dins la nostra xarxa local, per això modificarem arxiu /etc/network/interfaces

\$ sudo nano /etc/network/interfaces

Adreçament dinàmic amb dhcp

auto eth0

iface eth0 inet dhcp

Adreçament estàtic

auto eth0

iface eth0 inet static

address 192.168.1.10

netmask 255.255.255.0

gateway 192.168.1.1

- Ara reiniciarem la nova configuració de xarxa, amb el que perdrem la connexió de forma temporal.

\$ /etc/init.d/networking restart

* Reconfiguring network interfaces...

- Fixarem el nom del nostre servidor al /etc/hosts:
- D'aquesta manera hi podrem accedir sense haver d'invocar la IP local

\$ sudo nano /etc/hosts

Afegirem la línia següent:

192.168.1.10 celeron # servidor local

Comprovem que funciona

\$ ping celeron

PING celeron (192.168.1.10) 56(84) bytes of data.

64 bytes from celeron (192.168.1.10): icmp_req=1 ttl=64 time=9.37 ms

64 bytes from celeron (192.168.1.10): icmp_req=2 ttl=64 time=0.504 ms

- Tornarem a connectar desde el nostre ordinador

\$ ssh usuari@celeron -p 2222

The authenticity of host '[celeron]:2222 ([192.168.1.10]:2222)' can't be established.

RSA key fingerprint is 53:99:b2:ae:6f:e8:49:8f:96:ab:4a:09:dd:0f:5d:39.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '[celeron]:2222,[192.168.1.10]:2222' (RSA) to the list of known hosts.

usuari@celeron's password:

Linux celeron 2.6.35-22-generic-pae #33-Ubuntu SMP Sun Sep 19 22:14:14 UTC 2010 i686 GNU/Linux

Ubuntu 10.10

Welcome to Ubuntu!

* Documentation: <https://help.ubuntu.com/>

Last login: Mon Mar 14 12:12:12 2011 from 192.168.1.33

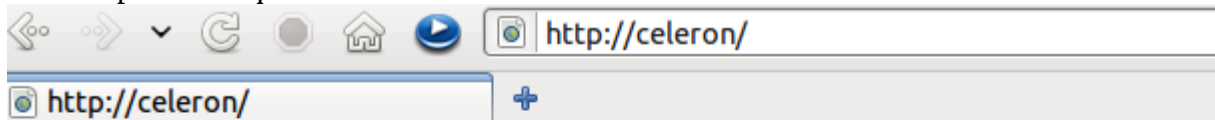
usuari@celeron:~\$

Instal·lació d'Apache2

- Instal·larem apache2:

```
$ sudo apt-get install apache2
```

- Comprovarem que funciona:



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Si ho creiem convenient modificarem els port per on s'escolta el servei web:

- Modificarem l'arxiu de ports per escoltar el 8080 enlloc del 80:

```
$ nano /etc/apache2/ports.conf
```

```
If you just change the port or add more ports here, you will likely also  
# have to change the VirtualHost statement in  
#/etc/apache2/sites-enabled/000-default  
# This is also true if you have upgraded from before 2.2.9-3 (i.e. from  
# Debian etch). See /usr/share/doc/apache2.2-common/NEWS.Debian.gz and  
# README.Debian.gz
```

```
NameVirtualHost *:8080
```

```
Listen 8080
```

```
<IfModule mod_ssl.c>
```

```
# If you add NameVirtualHost *:443 here, you will also have to change  
# the VirtualHost statement in /etc/apache2/sites-available/default-ssl  
# to <VirtualHost *:443>  
# Server Name Indication for SSL named virtual hosts is currently not
```

```
# supported by MSIE on Windows XP.  
Listen 443  
</IfModule>  
  
<IfModule mod_gnutls.c>  
Listen 443  
</IfModule>
```

- Opcionalment, podríem activar el servei de re-adreçament de ports (“port forwarding”) del router, per fer accessible el nostre servidor des d'Internet. Habilitarem a més un servei de DNS gratuït (noip2, en aquest cas cas).

Instal·lació MySQL5

- Instal·larem també el servidor de bases de dades MySQL(demana passwd) i comprovarem que funciona correctament amb:

```
$ sudo apt-get install mysql-server  
...  
$ sudo netstat -tap | grep mysql
```

- Canviarem l'arxiu /etc/mysql/my.cnf per escoltar peticions d'altres hosts de la xarxa:

Bind-address = 127.0.0.1 per bind-address = 192.168.1.10

Després, cal fer:

```
$ sudo /etc/init.d/mysql restart
```

Instal·lació PHP5

- Instal·larem PHP5 ...

Instal·lació phpMyAdmin

- Instal·larem phpMyAdmin per gestionar MySQL...

```
$ sudo apt-get install phpmyadmin
```

Fins aquí hem preparat un entorn 100% lliure per poder acomodar l nostra autoritat de certificació ara generarem els certificats necessaris, per crear la nostra pròpia CA.

Creació de la CA



El procés d'obtenció d'un certificat d'una CA és bastant senzill. La forma general ràpida seria la següent:

1. Crear un parell de xifrat de clau pública i privada.
2. Crear una sol·licitud de certificat sobre la base de la clau pública. La sol·licitud de certificat conté informació sobre el servidor i la companyia que l'allotja.
3. Enviar la sol·licitud de certificat, juntament amb documents que provin la seva identitat, a una CA (un cop ens hem decidit per una CA, cal seguir les instruccions que indiquin per obtenir el certificat per la nostra).
4. Quan l'autoritat competent estigui convençuda que nosaltres som qui diem ser, ens enviarà el certificat.
5. Instal·laríem aquest certificat en el servidor segur, i configurariem les aplicacions adequades per utilitzar el certificat.

En el nostre cas no seguirem els passos 3 i 4, sinó que crearem la nostra CA pròpia amb un certificat autosignat com seria el cas d'una CA arrel.

Generar una petició de signatura de certificat (CSR)



El primer pas és generar una clau. Si el certificat serà utilitzat pels dimonis de diferents serveis, com ara Apache, hauriem de crear una clau no protegida a amb contrasenya, ja que en iniciar el servei ens la demanaria cada vegada. El fet de no protegir la clau privada amb una contrasenya, és insegur i per tant pot comprometre tota la infraestructura de la CA, es per això que he optat per deixar-la fora de l'arbre del directori web del servidor. Aquesta consideració condiona el disseny posterior de l'aplicació web (a l'hora de signar i expedir certificats), aquest incís es tractarà en profunditat més endavant.

Per generar les claus per a la sol·licitud de signatura de certificat (CSR), executarem l'ordre següent des d'un terminal:

```
~$ openssl genrsa -des3 -out servidor.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
..++++++
e is 65537 (0x10001)
Enter pass phrase for servidor.key:
Verifying - Enter pass phrase for servidor.key:
~$
```

La contrasenya hauria de tenir un mínim de 4 caràcters per l'algorisme de xifrat (des3), encara que es recomana un mínim de 8 caràcters per considerar-la prou segura. Aquesta comanda ens generarà la clau privada: servidor.key xifrada amb triple-des amb la contrasenya escollida. Ara farem:

```
~$ openssl rsa -in servidor.key -out servidor.key.insegura
Enter pass phrase for servidor.key:
writing RSA key
~$ mv servidor.key servidor.key.segura
```



```
~$ mv servidor.key.insegura servidor.key
```

```
~$
```

Això crea una versió de la clau segura (xifrada) sense contrasenya a `servidor.key`, que podrà ser utilitzada pels diferents serveis. Si no ho féssim així cada cop que reiniciéssim el servidor ens demanaria la contrasenya de xifratge de la clau privada.

Per la nostra CA podem utilitzarem la clau privada xifrada:

```
~$ mv servidor.key.segura clauCA.key
```

```
~$ openssl req -new -key clauCA.key -out CertCA.csr
```

Enter pass phrase for `clauCA.key`:

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]:Catalunya

Locality Name (eg, city) []:Tarragona

Organization Name (eg, company) [Internet Widgits Pty Ltd]:TFC

Organizational Unit Name (eg, section) []:

Common Name (eg, YOUR name) []:Eugeni

Email Address []:eugeni@localhost

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password []:

An optional company name []:

```
~$
```

Ara ja tenim una CSR llesta per lliurar-la a una CA reconeguda o bé auto signar-la, com farem a continuació.

Generació d'un certificat autosignat



La següent comanda farà la feina:

```
~$ openssl x509 -req -days 365 -in CertCA.csr -signkey clauCA.key -out certCA.crt
Signature ok
subject=/C=ES/ST=Catalunya/L=Tarragona/O=TFC/CN=Eugeni/emailAddress=eugeni@localhost
Getting Private key
Enter pass phrase for clauCA.key:
~$
```

Instal·lació del certificat

El certificat i la clau de la CA les posarem a directoris segurs, fora de l'arbre web. Més endavant ja estudiarem la manera que els serveis de l'aplicació web de la CA accediran al certificat i la clau privada, però en principi intentarem ser el més reservats possible.

```
~$ sudo cp certCA.crt /etc/ssl/certs
[sudo] password for xxxx:
~$ sudo cp clauCA.key /etc/ssl/private/
~$
```

Creació i configuració de la nostra CA

- Crearem l'estructura de directoris per contenir els diferents certificats generats per la nostra CA i altres arxius relacionats:

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

- La CA necessita 2 fitxers addicionals per funcionar, un no perdre de vista l'últim número de sèrie (únic) utilitzat per la CA per identificar cada certificat, i un altre fitxer de registre pels certificats que han estat emesos:

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

- Per les funcionalitats que requerirà la nostra CA, necessitarem controlar els certificats revocats (crl) i els seus números de sèrie. Per això es creen directori d'arxius revocats i l'arxiu crlnumber (per controlar els números de sèrie):

```
# mkdir /etc/ssl/crl
# echo 01 > /etc/ssl/CA/crlnumber
#
```

- Ara editarem l'arxiu /etc/ssl/openssl.cnf . La secció corresponent a al CA per defecte ha de quedar com segueix:

```
[ CA_default ]

dir                = /etc/ssl                # Where everything is kept
certs              = $dir/certs              # Where the issued certs are kept
crl_dir            = $dir/crl                # Where the issued crl are kept
database           = $dir/CA/index.txt      # database index file.
#unique_subject    = no                     # Set to 'no' to allow creation of
# several ctificates with same subject.
new_certs_dir      = $dir/newcerts          # default place for new certs.

certificate        = $dir/certs/cacert.pem   # The CA certificate
serial             = $dir/CA/serial          # The current serial number
crlnumber          = $dir/CA/crlnumber       # the current crl number
# must be commented out to leave a V1 CRL

crl                = $dir/crl.pem            # The current CRL
private_key        = $dir/private/cakey.pem  # The private key
RANDFILE           = $dir/private/.rand     # private random number file

x509_extensions    = usr_cert               # The extentions to add to the cert
```

- Ara crearem i instal·larem el certificat arrel auto – signat:

```
~$ openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days 3650
Generating a 1024 bit RSA private key
.....++++++
...++++++
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]:Catalunya
Locality Name (eg, city) []:Tarragona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TFC
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:Eugeni
Email Address []:keni@localhost
~$ sudo mv cakey.pem /etc/ssl/private/
~$ sudo mv cacert.pem /etc/ssl/certs/
~$
```

Arribats a aquest punt podem començar a signar certificats i realitzar les tasques pròpies de la CA.

Operacions de la CA



Un cop instal·lada la CA al servidor, ja podem començar a executar les diferents operacions i serveis a través de la línia de comandes. Hem de tenir en compte però que aquests serveis (per requeriments de l'enunciat del TFC) s'han de poder accedir i ser gestionats a través d'una interfície web. Donat que hem decidit instal·lar la CA (la clau i el certificat) fora de l'arbre web per seguretat, construir la interfície en PHP i donat que les extensions de PHP per openssl no cobreixen tots els requeriments hem d'afrontar que els diferents serveis s'hauran d'executar al servidor a través de crides que realitzarà l'usuari web (www-data) mitjançant els diferents scripts PHP.

Està clar que haurem de donar certs privilegis i permisos a l'usuari web (www-data) per executar les comandes d'openssl, això ho farem mitjançant *sudo visudo*, més endavant i procurant no comprometre la seguretat del servidor.

El que farem ara és veure com es poden executar les diferents operacions de la CA al nostre servidor, comentarem en la mesura del possible cada paràmetre que utilitzem per poder integrar-lo amb la interfície web. Algunes d'aquestes funcionalitats seran:

1. **Generar peticions de certificat**
2. **Signar peticions de certificat**
3. **Mostrar un certificat**
4. **Veure període de validesa d'un certificat**
5. **Mostrar emissor d'un certificat**
6. **Mostrar propietari d'un certificat en format RFC2253**
7. **Revocar certificats**
8. **Generar la llista de certificats revocats**
9. **Llistar els certificats revocats**
10. **Mostrar si un certificat està revocat**
11. **Renovar un certificat**

Generar peticions de certificat

Generem la clau del client xifrada amb contrasenya (donada per l'usuari) i després la petició de certificació amb les dades també proporcionades per l'usuari. Desarem la clau de l'usuari i la petició de l'usuari als directoris que hem creat abans.

```
~$ openssl genrsa -des3 -passout pass:contrasenya -out /tmp/ususari.key
Generating RSA private key, 512 bit long modulus
.....+++++++
.....+++++++
e is 65537 (0x10001)
~$
```

```
~$ openssl req -new -key /tmp/ususari.key -passin pass:contrasenya -subj
'/CN=usuari/O=NomOrganitzacio/C=ES/ST=Catalunya/L=Poblacio' -out /tmp/usuari.csr
~$
```

Les claus privades dels clients, en principi NO s'han de desar al servidor, és informació molt compromesa de desar, de fet es desaran temporalment a la BD de la CA i s'eliminaran un cop s'hagin lliurat de forma segura al client, juntament amb el certificat corresponent degudament signat.

Signar peticions de certificat

```
~$ sudo openssl ca -in /tmp/usuari.csr -batch -passin pass:contrasenyaCA -config /etc/ssl/openssl.cnf
-policy policy_anything
```

La opció -batch permet executar la comanda sense haver de realitzar les 2 confirmacions “yes” la opció policy_anything evita els errors de no – coincidències en els noms distintius de l'usuari i la CA.

Revocar certificats

```
~$ sudo openssl ca -revoke /etc/ssl/newcerts/09.pem -passin pass:passwdCA -config /etc/ssl/openssl.cnf
Using configuration from /etc/ssl/openssl.cnf
Revoking Certificate 09.
Data Base Updated
~$
```

Generar la llista de certificats revocats

```
~$ sudo openssl ca -genctrl -out /etc/ssl/crl/revocats.crl -config /etc/ssl/openssl.cnf -passin pass:pasCA
Using configuration from /etc/ssl/openssl.cnf
~$
```

Mostrar la llista de revocats

```
~$ sudo openssl crl -in /etc/ssl/crl/revocats.crl -text
Certificate Revocation List (CRL):
  Version 2 (0x1)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: /C=ES/ST=Catalunya/L=Tarragona/O=TFC/CN=Eugeni/emailAddress=keni@localhost
  Last Update: Apr 11 18:23:53 2011 GMT
  Next Update: May 11 18:23:53 2011 GMT
  CRL extensions:
    X509v3 CRL Number:
      1
Revoked Certificates:
  Serial Number: 07
    Revocation Date: Apr 11 18:18:17 2011 GMT
  Serial Number: 08
    Revocation Date: Apr 11 18:14:35 2011 GMT
  Serial Number: 09
    Revocation Date: Apr 11 18:11:09 2011 GMT
  Signature Algorithm: sha1WithRSAEncryption
  63:65:36:f6:21:83:7f:4b:0f:d4:93:e1:25:de:20:10:c1:cd:
  a5:03:6c:31:5f:cf:60:c6:0d:34:b0:f4:6a:25:6a:58:af:dd:
  b1:52:33:0f:7d:bb:5d:90:93:3f:1e:07:07:20:e8:4c:4c:da:
  79:d6:49:cc:d7:50:df:be:18:aa:dc:7c:5c:71:68:7e:45:43:
  c8:8f:dd:24:b7:f2:1b:38:59:f0:84:66:9e:f4:61:a5:ee:c5:
  76:47:5b:86:86:38:d5:17:f5:22:d9:44:26:2d:db:d3:39:cb:
  2a:2c:9d:28:18:ae:07:ca:74:b4:5b:82:10:53:1f:60:5f:a4:
  ac:95
-----BEGIN X509 CRL-----
MIIBiTCB8wIBATANBgkqhkiG9w0BAQUFADBzMQswCQYDVQQGEwJFUzESMBAGA1UE
CBMJQ2F0YWx1bnlhMRIwEAYDVQQHEwlUYXJyYWdvbmExDDAKBgNVBAoTA1RGQzEP
MA0GA1UEAxMGRXVnZW5pMR0wGwYJKoZIhvcNAQkBFg5rZW5pQGxvY2FsaG9zdBcN
MTEwNDExMTgyMzUzWhcNMTEwNTEwMTgyMzUzWjA8MBICAQcXDTEwMDQxMTE4MTgx
N1owEgIBCBcNMTEwNDExMTgxNDM1WjASAgEJFw0xMTA0MTEwODEwMDlaoA4wDDAK
BgNVHRQEAwIBATANBgkqhkiG9w0BAQUFAAQBjZTb2IYN/Sw/Uk+El3iAQwc2l
A2wxX89gXg00sPRqJWpYr92xUjMPfbdtkJM/HgcHIOhMTNp51knM11Dfvhiq3Hxc
cWh+RUPiJ90kt/IbOFnwhGae9GGI7sV2R1uGhjjVF/Ui2UQmLdvTOcsqLJ0oGK4H
ynS0W4IQUx9gX6SslQ==
-----END X509 CRL-----
```

La CRL final es publicarà posteriorment en la forma adequada.

Mostrar si un certificat està revocat



```
~$ openssl ca -status 01 -config /etc/ssl/openssl.cnf
Using configuration from /etc/ssl/openssl.cnf
01=Valid (V)
~$ openssl ca -status 08 -config /etc/ssl/openssl.cnf
Using configuration from /etc/ssl/openssl.cnf
08=Revoked (R)
~$
```

Renovar un certificat



```
~$ sudo openssl ca -config /etc/ssl/openssl.cnf -policy policy_anything -passin pass:pasCA -batch
-infiles /tmp/usuari9.csr
```

A partir de la mateixa petició que havia fet l'usuari, tornem a generar un nou certificat, també es pot processar una nova petició signada amb la clau de l'usuari.

Mostrar propietari



```
~$ openssl x509 -noout -in /etc/ssl/newcerts/0A.pem -subject
subject= /C=ES/ST=Catalunya/L=Poblacio/O=NomOrganitzacio/CN=usuari9
~$
```


Configuració de l' HTTPS

La nostra aplicació ha de funcionar sota la capa ssl a través del protocol https, per activar aquesta característica del servidor apache2 utilitzarem el certificat i la clau privada de la CA que hem creat anteriorment. Per fer això seguirem els següents passos:

- Activem el mòdul ssl del l'apache2

```
keni@phenom:~$ sudo a2enmod ssl
[sudo] password for keni:
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
create self-signed certificates.
Run '/etc/init.d/apache2 restart' to activate new configuration!
keni@phenom:~$
```

- Editem l'arxiu de configuració de l'HTTPS /etc/apache2/sites-available/default-ssl . Indiquem la clau privada i el certificat **de la nostra CA**, que hem creat.

```
# Enable/disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by inst
# the ssl-cert package. See
# /usr/share/doc/apache2.2-common/README.Debian.gz for more i
# If both key and certificate are stored in the same file, on
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/cacert.pem
SSLCertificateKeyFile /etc/ssl/private/cakey.pem
```

- Reiniciem el servei apache2.

```
keni@phenom:~/etc/apache2/sites-available$ sudo nano default-ssl
keni@phenom:/etc/apache2/sites-available$ sudo a2ensite default-ssl
Site default-ssl already enabled
keni@phenom:/etc/apache2/sites-available$ sudo /etc/init.d/apache2 restart
* Restarting web server apache2
... waiting Apache/2.2.16 mod_ssl/2.2.16 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server localhost.localdomain:443 (RSA)
Enter pass phrase:
```

- Si ens demana la contrasenya de la clau privada de la nostra CA, l'hauem de posar ara:

```
In order to read them you have to provide the pass phrases.
```

```
Server localhost.localdomain:443 (RSA)
```

```
Enter pass phrase:
```

```
OK: Pass Phrase Dialog successful.
```

```
keni@phenom:/etc/apache2/sites-available$ █
```

1.2.1. Basic Settings

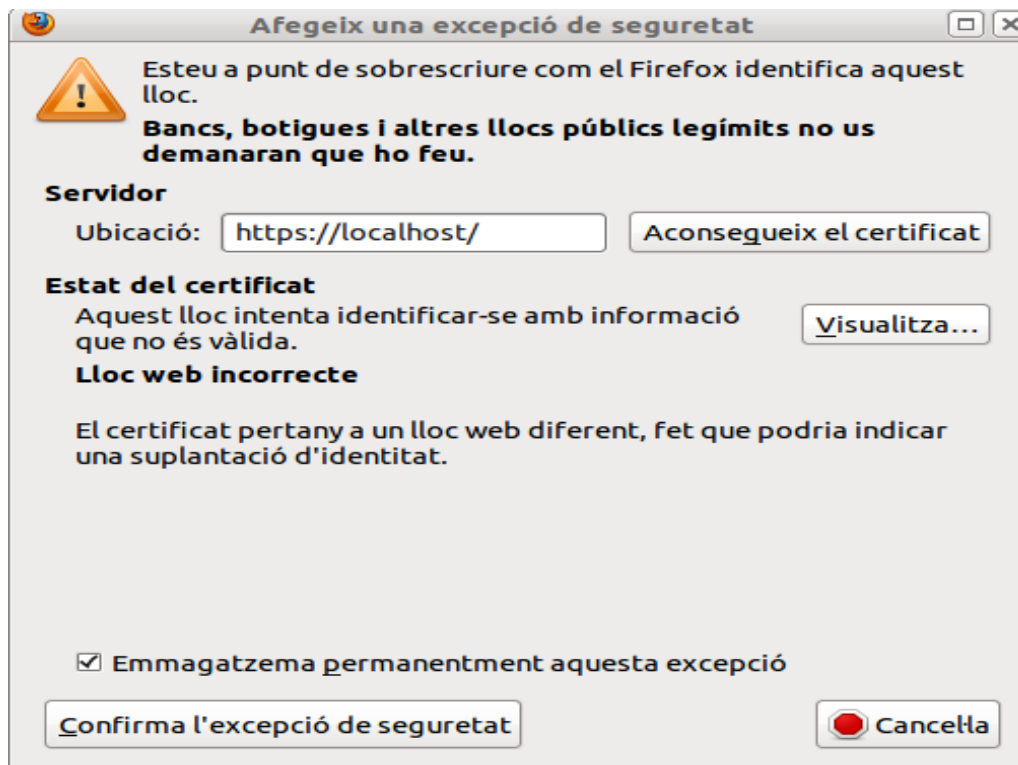
141

- I ja està, si accedim des del navegador al directori web segur, ens demanarà l'acceptació del nostre certificat auto-signat (que és considerat com insegur) i acceptarem, ja que aquí ja sabem el que estem fent.

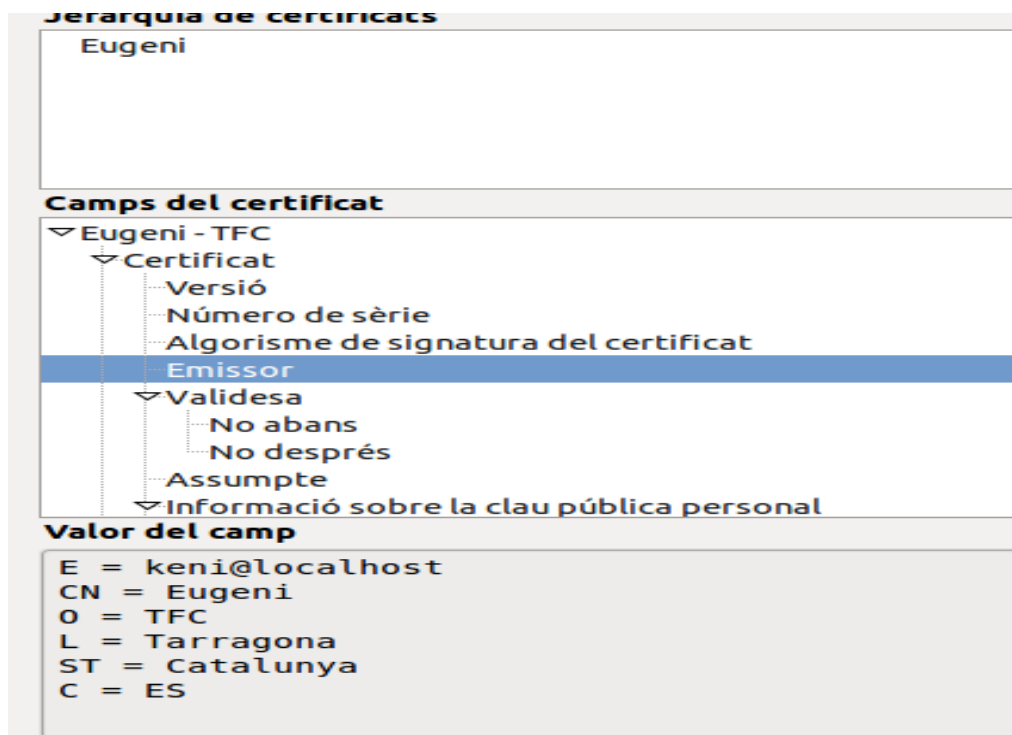


The screenshot shows a Firefox browser window with the address bar containing `https://localhost/tfc/`. A yellow warning box is displayed in the center of the page. The warning box has a yellow background and a black border. It contains a yellow icon of a person with a question mark, followed by the heading **No es pot confiar en la connexió**. Below the heading, there is a paragraph of text: "Heu demanat al Firefox connectar-se de forma segura a **localhost**, però no podem confirmar que la vostra connexió ho sigui." This is followed by another paragraph: "Normalment, quan intenteu connectar-vos de forma segura, els llocs web us presentaran una identificació pertinent per a demostrar-vos que aneu al lloc adequat. No obstant això, la identitat d'aquest lloc no es pot verificar." Below this is the heading **Què caldria que fes?** and a paragraph: "Si normalment us connecteu a aquest lloc sense problemes, podria voler dir que algú altre vol fer-se'n amb la identitat i, per tant, no hauríeu de continuar." At the bottom of the warning box, there is a button labeled "Treu-me d'ací!" and two links: **▶ Detalls tècnics** and **▶ Entenc els riscos**.

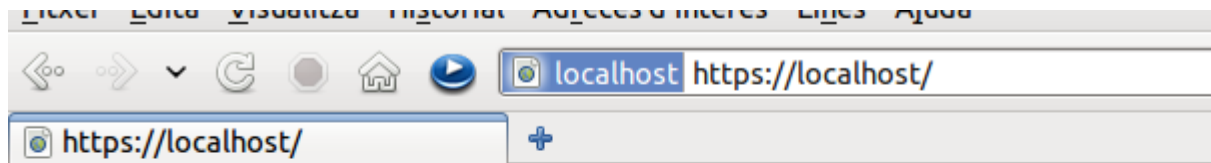
- El navegador intenta fer-nos por (i fa bé!) abans d'acceptar el certificat.



- Si confirmen la excepció i mirem els detalls podrem verificar que es tracta del nostre certificat auto-signat, tranquils i endavant.



- Si posem ara l'adreça web al nostre navegador:



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

- Ara habilitem el lloc on anirà el nostre projecte (/var/www/tfc/)modificant l'arxiu /etc/apache2/sites-available/default-ssl per que quedi així:

```
GNU nano 2.2.4                               Fitxer: default-ssl
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/tfc/
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/tfc/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```

Permissos per l'usuari web (www-data)



Les comandes de la CA, tal com ho hem pensat en principi, s'han d'executar desde la web on residirà l'aplicació. Algunes d'aquestes comandes necessiten accedir a recursos protegits i d'accés restringit. Les extensions de PHP per openssl no són del tot suficients per la nostra CA, especialment en la gestió dels certificats revocats. Per tant hem de dotar de privilegis (alguns, no tots) a l'usuari web, la manera de fer això s'aconsegueix en GNU/Linux mitjançant el comandament visudo:

```
~$ sudo visudo
```

Editem la secció de privilegis per que quedi de la següent forma:

```
# User privilege specification
root ALL=(ALL) ALL
www-data ALL=NOPASSWD: /usr/bin/openssl ca + paràmetres que considerem
www-data ALL=NOPASSWD: /usr/bin/openssl req + paràmetres que considerem
```

Aquest és el punt més compromès de la CA que em dissenyat, el fet de dotar de privilegis de super-usuari a l'usuari web (www-data en el nostre cas) es dona a rebre atacs, que podem pagar molt cars. Malgrat aquest fet, podem restringir l'execució de les comandes fent que només es puguin executar amb determinats paràmetres i opcions, evitant així una possible situació d'abús.

De fet dins l'aplicació es donen les següents execucions amb privilegis de super-usuari

Per signar les peticions:

```
exec ("sudo openssl ca -in /tmp/peticio.csr -batch -passin pass:$passca -config /etc/ssl/openssl.cnf -policy policy_anything -out /tmp/cert.pem -notext 2>&1", $revs, $retorna);
```

Per revocar certificats:

```
exec ("sudo openssl ca -revoke /tmp/cert.pem -passin passpassca -config /etc/ssl/openssl.cnf 2>&1", $revs, $retorna);
```

Per generar la llista de revocació:

```
exec("sudo openssl ca -gencrl -out /etc/ssl/crl/revocats.crl -config /etc/ssl/openssl.cnf -passin pass:$pwdCA 2>&1", $revs, $retorna);
```

Es pot acotar mitjançant visudo l'execució d'aquestes comandes restringint a més del executable els paràmetres que porta. La sortida de openssl s'ha de re-dirigir cap a la sortida estàndard (**2>&1**) i recollir en una matriu (**\$revs**) el resultat de la comanda, el paràmetre **\$retorna**, serà 1("status") si es produeix algun error.

Malgrat les precaucions que puguem prendre, hauria estat preferible utilitzar les extensions de php5 per openssl, però també m'ha servit per aprendre aspectes de seguretat importants i útils.

Aplicació web



Bàsicament presenta un disseny senzill on es distingeix una àrea de serveis destinada a expedir certificats, i una secció d'administració per signar, revocar certificats i la generació de llistes de revocació

Per tal accedir als serveis de la CA cal autenticar-se després de registrar-se com a usuari a la BD les contrasenyes es desen xifrades i només l'usuari es serà el coneixedor.

Per tal de realitzar tasques administratives cal també autenticar-se coma administrador de la CA, en aquest cas també es desa la contrasenya xifrada i no ha de coincidir necessàriament amb la paraula clau de xifratge de la clau privada de la CA.

La navegació pels diferents serveis, tant dels usuari com de l'administrador, es realitza a través de sessions on l'usuari o l'administrador han d'estar autenticats. Si no, no es permetrà l'accés al servei.

A continuació, sense fer una explicació exhaustiva de les diferents classes i mètodes dels mòduls de programari (que ja estan comentats en el codi font proporcionat) es fa una exposició de les funcionalitats implementades:

Inici i registre de clients



TFC Safe Key CA

Inici Contacte Serveis Administrador Ajuda

CA SK, és una autoritat de certificació creada per donar una sèrie de serveis de signatura de certificats i autenticació en línia

TFC Safe Key CA Ltd.
This work is licensed under a [Creative Commons Attribution 3.0 Unported License](#).

Captura 1: Aspecte inicial de l'entorn

TFC Safe Key CA

Inici Contacte Serveis Administrador Ajuda

Generar Clau i Petició de certificat Introduir Petició de certificat Emissió de certificat Llista de certificats revocats Solicitud de Revocació Nou usuari Sortir

Cal identificar-se com a usuari registrat

Identificació d'usuari

Usuari

Contrasenya

ok

TFC Safe Key CA Ltd.

Captura 2: Autenticació de l'usuari

Un cop autenticats s'inicia la sessió d'usuari que ens permet accedir als diferents serveis, aquesta sessió finalitzarà, quan l'usuari vagi a "sortir" i ho confirmi .

Captura 3: Registre d'un nou usuari

Si no estem autenticats a la BD hem de registrar-nos amb un senzill formulari.

Generar Clau i Petició de certificat: Es genera una clau privada en format PEM que es desarà xifrada a la BD. Amb les dades del formulari i la clau generada, es crea una petició de certificat que també es desa a la BD. La petició quedarà a l'espera de ser signada per la CA.

Introduir Petició de certificat: L'usuari ha generat la seva pròpia petició de certificat en format PEM. Es demana enganxar la petició pkcs#10 a un quadre de text. S'extreuen les dades necessàries i s'actualitza la BD. No cal facilitar la clau privada de l'usuari però si es vol importar el certificat en format pkcs#12, caldrà proporcionar-ho més endavant. La petició quedarà a l'espera de ser signada per la CA.

Llista de certificats revocats: Un cop autenticat pot accedir a la llista de certificats revocats de la CA.

Solicitud de Revocació: L'usuari ha d'estar registrat a la BD de certificats signats. Un cop autenticat pot sol·licitar la revocació del seu certificat a la CA.

Sortir: Sortir de la sessió d'usuari.

Captura 4: Informació inicial dels serveis

El primer cop que iniciem la sessió d'usuari se'ns presenta informació resumida sobre els diferents serveis.

Captura 5: Finalitza sessió usuari

Peticions de signatura

[Inici](#)[Contacte](#)[Serveis](#)[Administrador](#)

Informació de l'usuari

Aquesta informació es mostrarà al certificat emès.

Nom Complert(*)

Correu Electrònic(*)

Organització(*)

Department

Situació Geogràfica

Població(*)

Província

País(*)

Contrasenya(*)

(*) Camps obligatoris

Captura 6: Genara la petició de signatura

Un cop autenticats, poden realitzar peticions de signatura de certificat omplint el formulari de la captura 6. Això generarà una clau privada xifrada amb la contrasenya de l'usuari, que es desarà temporalment a la BD fins a l'expedició del certificat PKCS#12. La petició quedarà pendent de ser signada per la CA (administrador).



Les dades ha estat ben introduïdes:

Antònia Corriola
Morcillas SA
SPAIN
Tarragona
Vilacodonys
Departamento Choricero
keni@localhost.local.lan

S'ha enviat un codi secret al correu proporcionat, introduïu-lo al següent cuadro de text.

Validació per Email

Introduïu el text xifrat aquí:

Secret

Captura 7: Validació del correu proporcionat al realitzar la petició

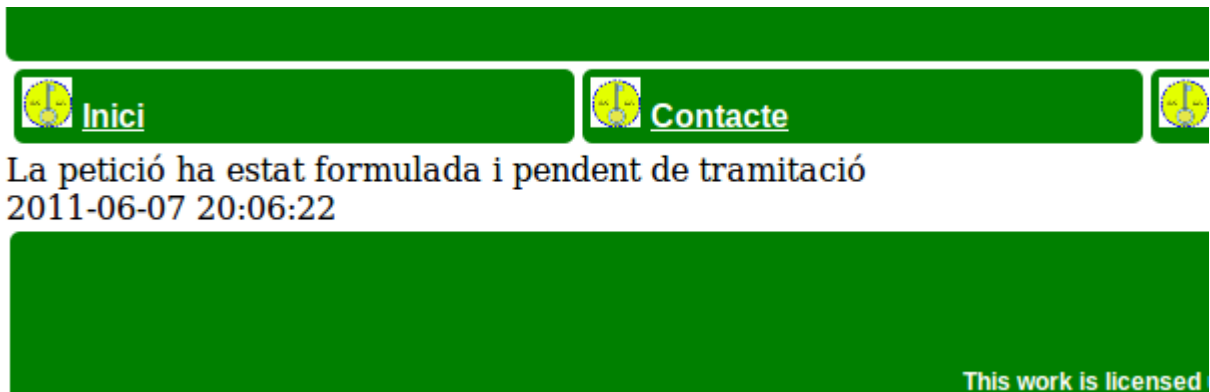
Si es realitza la petició de signatura correctament s'envia un correu a l'adreça proporcionada per l'usuari. En aquest correu hi ha un missatge a copiar en el requadre, validant així el correu proporcionat. Això NO representa una validació exhaustiva que és competència d'una autoritat de validació. La CA estrictament parlant expedeix certificats però la validació es pot delegar a un altre organisme (que pot ser la mateixa CA).

```
Return-Path: <www-data@phenom.local.lan>
X-Original-To: keni@localhost.local.lan
Delivered-To: keni@localhost.local.lan
Received: by phenom (Postfix, from userid 33)
        id C159DB816BC; Tue,  7 Jun 2011 20:27:34 +0200 (CEST)
To: keni@localhost.local.lan
Subject: Missatge secret de TFC CA
X-PHP-Originating-Script: 1000:csr1.php
Message-Id: <20110607182734.C159DB816BC@phenom>
Date: Tue,  7 Jun 2011 20:27:34 +0200 (CEST)
From: www-data@phenom.local.lan (www-data)
```

```
Hola Antònia Corriola,<br>Aquest és un correu pel tal de
confirmar l'adreça que has proporcionat al formulari de petició de
certificat.Copia el següent codi i enganxa'l on es demana a la pagina
d'origen: 725339613dc9e7ce990b9a6d898dbb39d0e3ec00<br>Salutacions,
<br><br> TFC CA
```

Captura 8: Correu lliurat amb la paraula secreta

Si tot va bé, aleshores quedarem assabentats amb un missatge:



Captura 9: Petició acceptada i pendent de signatura

Signatura de peticions



[Inici](#)
[Contacte](#)
[Serveis](#)
[Administrador](#)

[Signar peticions](#)
[Revocar certificats](#)
[Requeses](#)

Cal autenticar-se com administrador

Contrasenya de l'administrador

Contrasenya_Adm

Captura 10: Autenticació de l'administrador

Ens autenticarem com a administrador per tal de veure les peticions de signatura.

TFC Safe Key CA

[Inici](#)
[Contacte](#)
[Serveis](#)
[Administrador](#)

Peticions pendents de signar

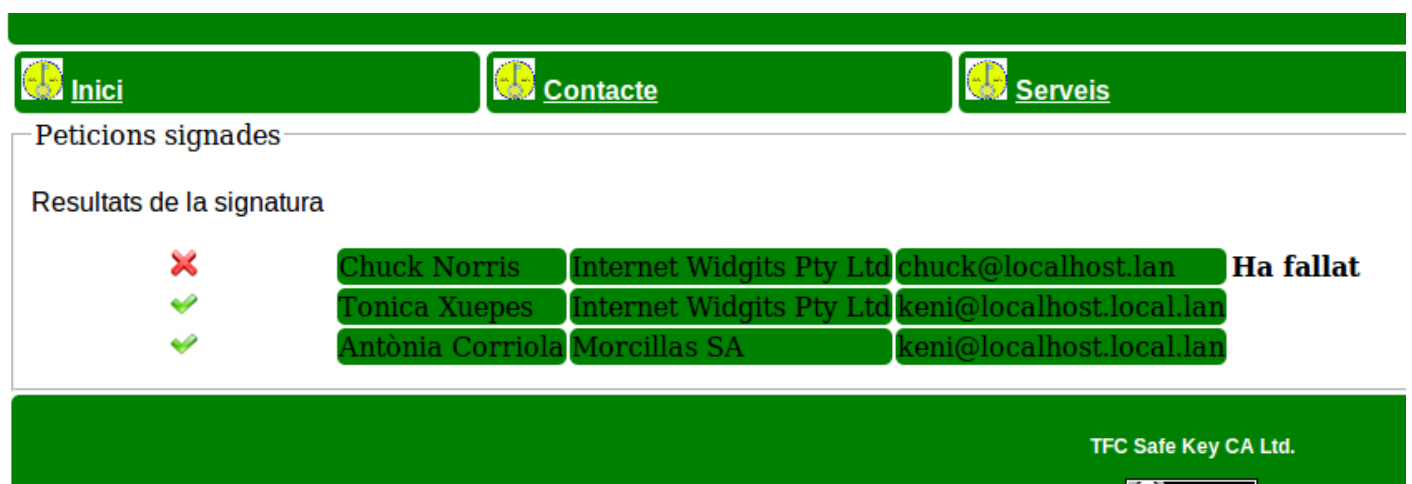
Hi ha 4 peticions pendents. Seleccionar les que es puguin signar.

<input checked="" type="checkbox"/>	Chuck Norris	Internet Widgits Pty Ltd	chuck@localhost.lan
<input type="checkbox"/>	Bob Cacaolet	Internet Widgits Pty Ltd	keni@localhost.local.lan
<input checked="" type="checkbox"/>	Tonica Xuepes	Internet Widgits Pty Ltd	keni@localhost.local.lan
<input checked="" type="checkbox"/>	Antonia Corriola	Morcillas SA	keni@localhost.local.lan

TFC Safe Key CA Ltd.

Captura 11: Signar les peticions pendents

Ara podem seleccionar quines peticions signem i veure si han tingut èxit.



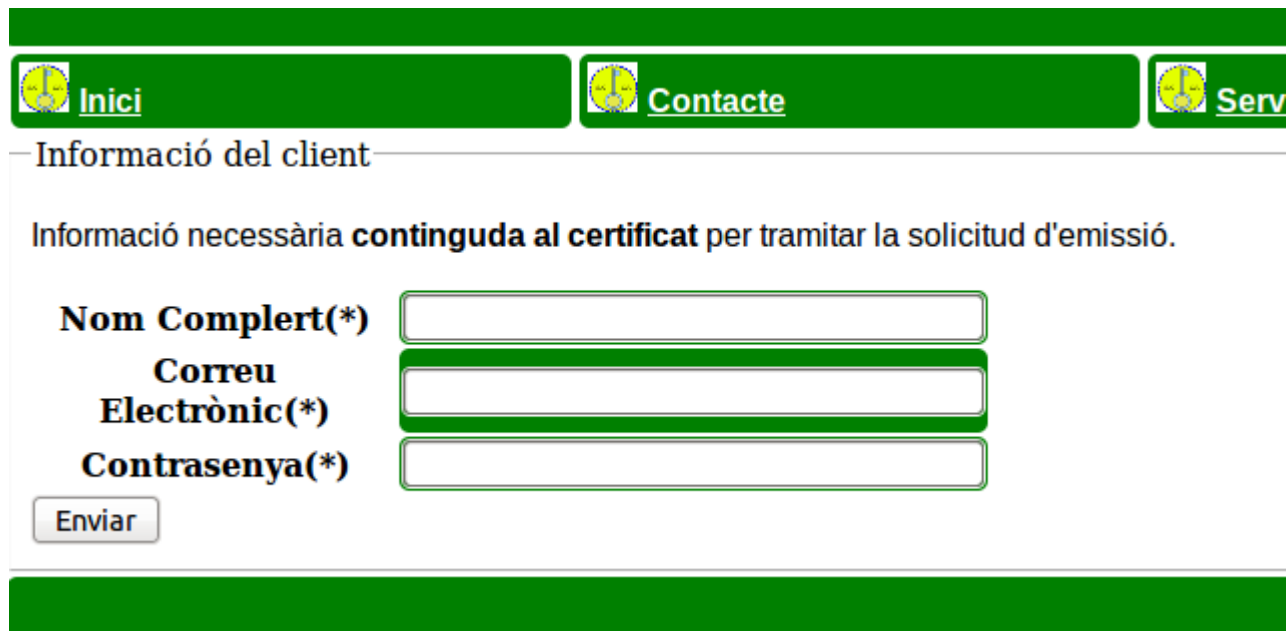
The screenshot shows a navigation bar with three buttons: 'Inici', 'Contacte', and 'Serveis'. Below the navigation bar, the text 'Peticions signades' is followed by 'Resultats de la signatura'. A table lists three entries with their respective status icons (red X, green checkmarks) and details. The first entry, 'Chuck Norris', is marked as 'Ha fallat' (has failed). The second and third entries, 'Tonica Xuepes' and 'Antònia Corriola', are marked as successful. The footer of the page reads 'TFC Safe Key CA Ltd.'.

Icona	Nom	Entitat	Correu	Resultat
✗	Chuck Norris	Internet Widgits Pty Ltd	chuck@localhost.lan	Ha fallat
✓	Tonica Xuepes	Internet Widgits Pty Ltd	keni@localhost.local.lan	Correcte
✓	Antònia Corriola	Morcillas SA	keni@localhost.local.lan	Correcte

Captura 12: Resultat de la signatura

Emissió de certificats PKCS#12

A la sessió d'usuari ara podem accedir a la secció d'expedició de certificats. Per demanar el certificat corresponent, es demana introduir algunes dades contingudes al mateix, per tal d'identificar-lo de forma unívoca.



The screenshot shows a navigation bar with three buttons: 'Inici', 'Contacte', and 'Serv'. Below the navigation bar, the text 'Informació del client' is followed by 'Informació necessària continguda al certificat per tramitar la sol·licitud d'emissió.'. There are three input fields for 'Nom Complert(*)', 'Correu Electrònic(*)', and 'Contrasenya(*)'. An 'Enviar' button is located below the input fields. The footer of the page is a solid green bar.

Captura 13: Sol·licitud d'emissió



El certificat està present a la BD:

Antònia Corriola
Morcillas SA
Departamento Choricero
keni@localhost.local.lan

Emissió de certificat

- Si es vol emetre el certificat en format PKCS#12, cal enganxar (si no ho està) al següent quadre de text La clau privada xifrada amb la contrasenya proporcionada.
- Si la clau privada es troba ja a la BD i es es vol eliminar després d'emetre el certificat seleccionar el quadre corresponent.

Certificat PEM:

```
-----BEGIN CERTIFICATE-----
MIICITCCAjGgAwIBAgIBKTBANBgkqhkiG9w0BAQUFAADBzMQswCQYDVQQGEwJFUzES
MBAGAlUECBMjQ2F0YXN1bn1hMFRlYAYDVQQHEwLUYyJyYwdWVibmExDDAKBgNVBAoT
A1RGRzQEPMAOGAlUEAaMRXVnZW5wMR0wGyYKozIhvcNAQBF5rZwSpQ0xvY2Fs
aG9zdDaeFw0xMTA2MDCxOTA3MzhaFw0xMjA2MDYxOTA3MzhaMIGOMQswCQYDVQQG
EwJFUzESMBAGAlUECBMjQ2F0YXN1bn1hMFRlYAYDVQQHEwLUYyJyYwdWVibmEx
MEMGAlUEChMNTW9yY21sbGFzIFNBMR8wHQYDVQQLExZEEZXBhcnRhbWVudG8gQ2hv
cm1jZXJvMR0wGAYDVQQDFBFbnTDSm5pYSDb3JyaW9sYENMCUGCSGSIb3DQEJ
ARYYa2VuaUBsb2NhbGhvc3QubG9jYyYwubG9wMFw0YyYKozIhvcNAQEBBQADSwAw
SAJBAMIEjBFWgZamtkUk2t6wcTL/wqE9uv+8t35xj4L9FQq0fPNZFsRwrBQh8I1
P-Avs9xWjYpCbqSbMF341Dbd00CAwEAAn7MhwCQYDVROTBAlWADAsBg1ghkG8
hvhCAQoEhYjT3B1b1NTTCBzH2H5LcaF0ZkQg22ydgLaAmhndGUm4YDVROBGEYE
FJXtB6zt1iabbQPZ40GV/PuipafmMB8GAlUdIwQYBAFIUZknZtTf6gVTzvUCT9
qNwjinpSMAOGCSGSIb3DQEBAQAAGBAbdyVUTSEqIRXt19zD06Q3CgMUDV189
Tawc1UgJxvSkpSh6t2jfelR4VYztkg21Gyqi+8T2reDQRj2wPcr43GEwIthSsOL
y/ZiUfBVS4LyyjPjW27Q1J67WfyIkuYpL1XtgXmrMknGaMNEuL0LXW152uHDE6B
```

Clau privada
xifrada:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC, E6342CE55042FC12
oCJt156aJ5MI0xrkHQ5ZQ2YEOZkTSUFegMnImcsPB5tW77P07obHUK9JLausbpV
fDA10owB9+/0jbsRBXVESegc4pYG7mU2BA6DIvCUKJvfgLms61G1tY3uSCT5E0a1
dyHhtr00UcNn7jL0iik6KkqCQFEwp/7atnS7tR+kpbdd20YjzTmQA1GOLDMabcX0x
0Z35R005cZiXeUxq4j9PvDtQwc+y29gg+tkWUWESNH3F/SpXZ/gJQ+/PsDEKYB1
7McrrpDgP3zrbULIEYmRnQ08d6nkKc5Y1FEL5iRnzRBTorWCEAutK1w1vE6Cuj
0wFUDStXPU5KQW553tgbjV1vRBXj/LoVlpu6X01mkF+2Vc0+0XQ00DBAK/Pg1Ty
UJRL/3x9AVrFLH1z/1Cpx9+GyUq842ZJRuv5g3vuHw=
-----END RSA PRIVATE KEY-----
```

Esborrar clau privada de la BD

Generar certificat PKCS#12

Captura 14: Emissió de certificat PEM

Si s'ha emès correctament el certificat, es mostra 2 quadres de text amb el certificat signat en format PEM i la clau privada del client xifrada amb la contrasenya del client, aquesta es va generar en el moment de realitzar la petició i es va desar a la BD.

El següent pas és l'emissió del certificat en format PKCS#12 corresponent i a partir d'aquí, no es necessita la clau privada del client (i es preferible no tenir-la). Per tant es proposa esborrar-la de la BD, marcant un quadre de selecció a sota.

Després de generar el certificat PKCS#12, es proposa la descàrrega dels certificats (PEM i PKCS#12) juntament amb la clau privada, que repeteixo serà ja esborrada de la BD.

Captura 15: Descàrrega de certificats i clau privada del client



Emissió de certificat

- El certificat s'ha exportat correctament a l'arxiu al format corresponent PKCS #12.
- Pots descarregar el arxius i i tornar a la pàgina de serveis.
- Si has seleccionat esborrar la clau privada de la BD, pots descarregar-la abans.

Certificat PEM:



Clau privada
xifrada:



Certificat
PKCS#12:

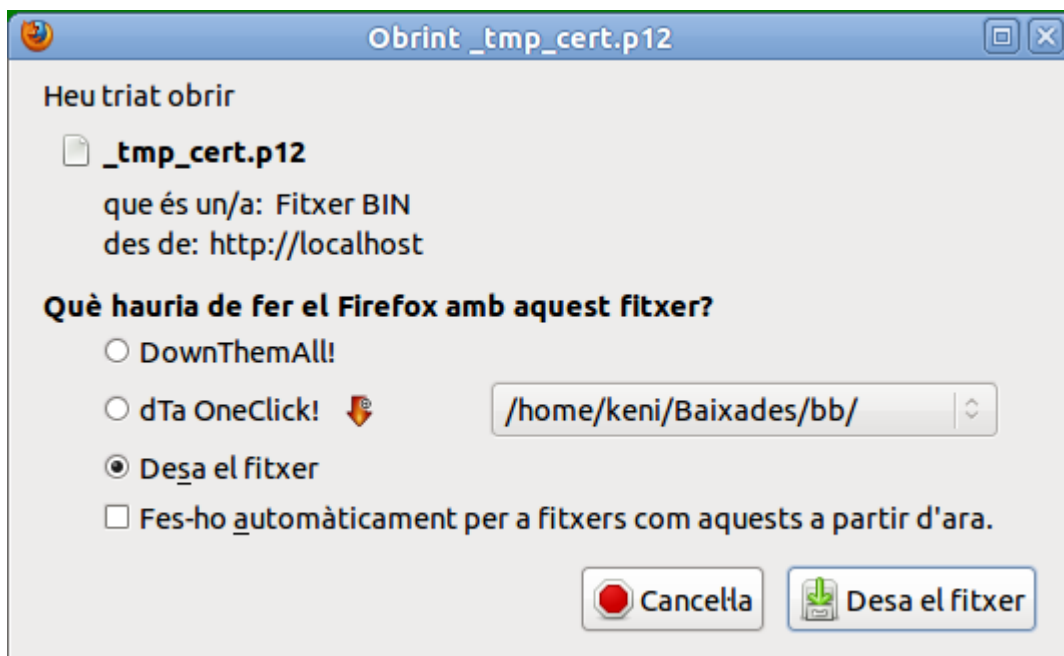


Ok

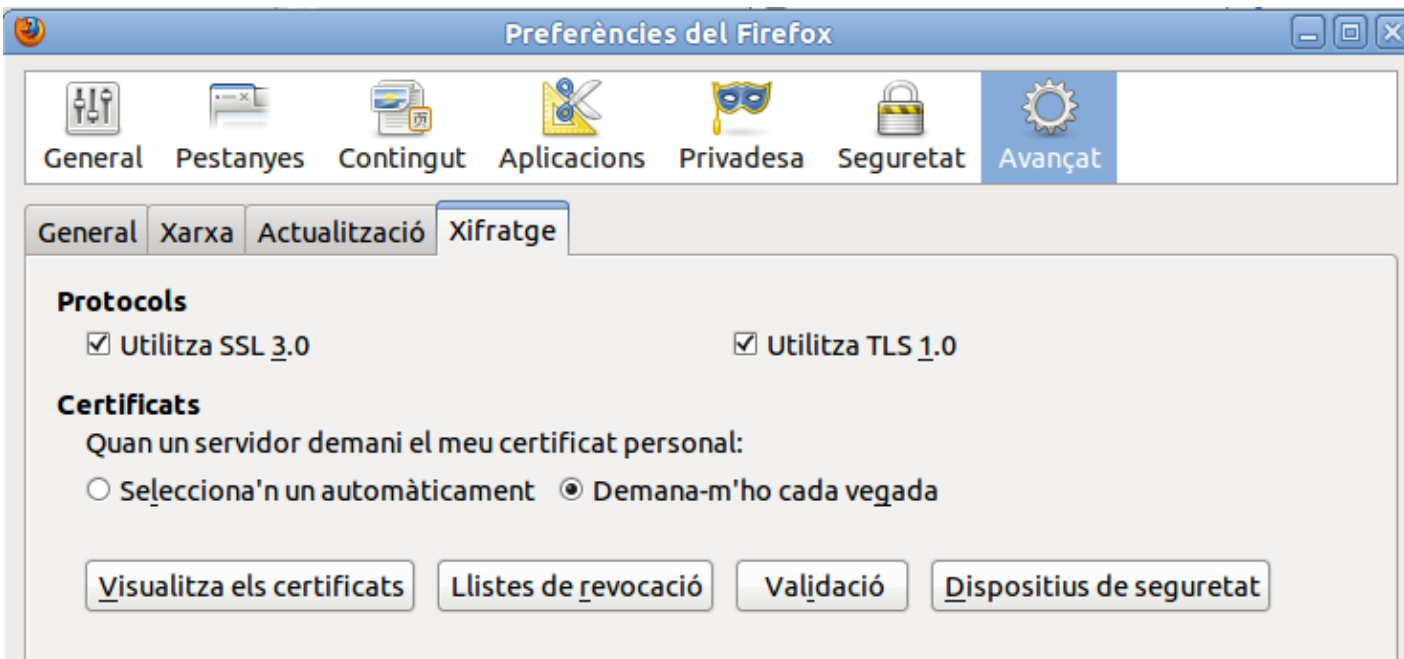
Si el client descarrega la seva clau privada podrà tornar a sol·licitar un altre certificat però si no ho fa en aquest moment, ja no la podrà recuperar, ja que serà esborrada just després de prémer el botó "Ok".

Si descarrega el certificat PKCS#12, pot instal·lar-lo al navegador. En en el meu cas (Firefox) seria com es mostra a continuació:

Descarreguem

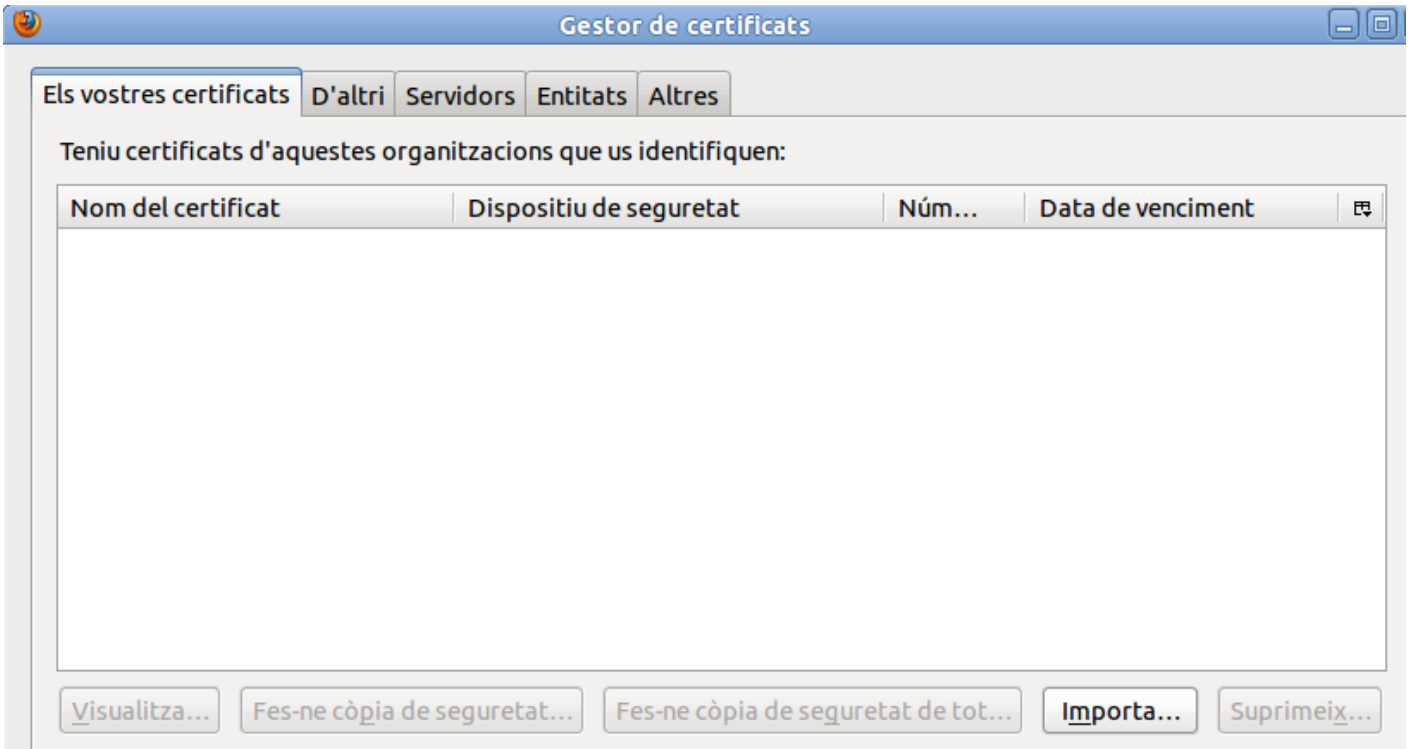


Captura 16: Descàrrega PKCS#12

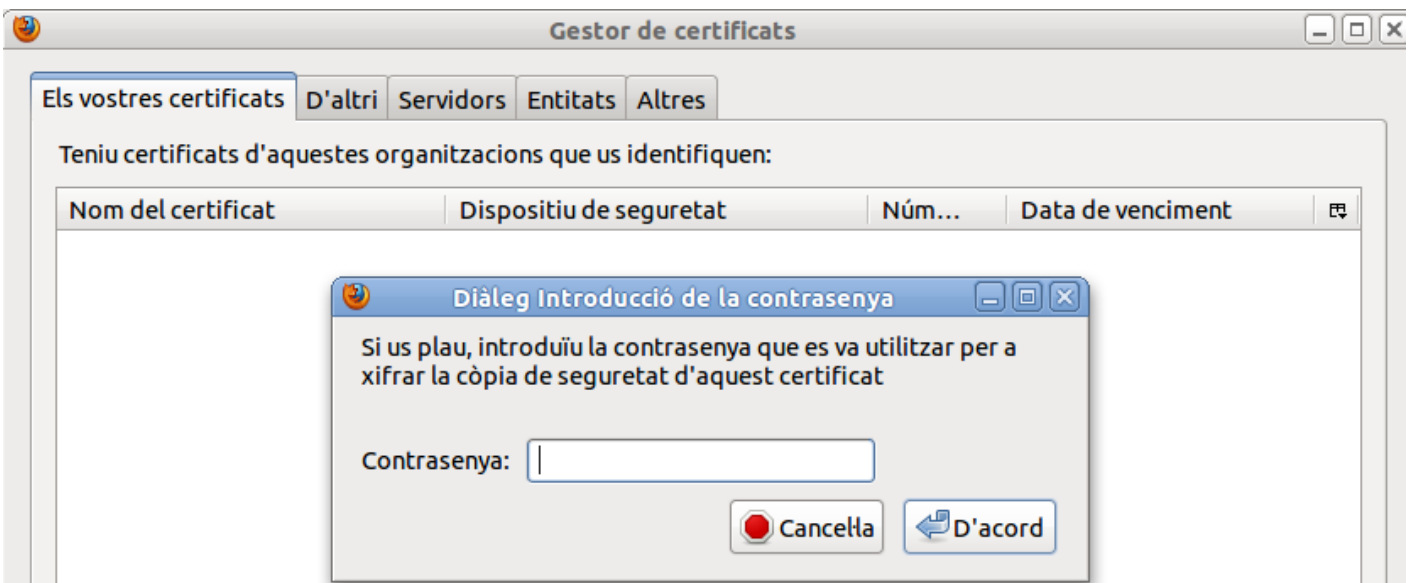


Captura 17: Preferències del Firefox, menu avançat

Anem a *visualitza els certificats / importa...*

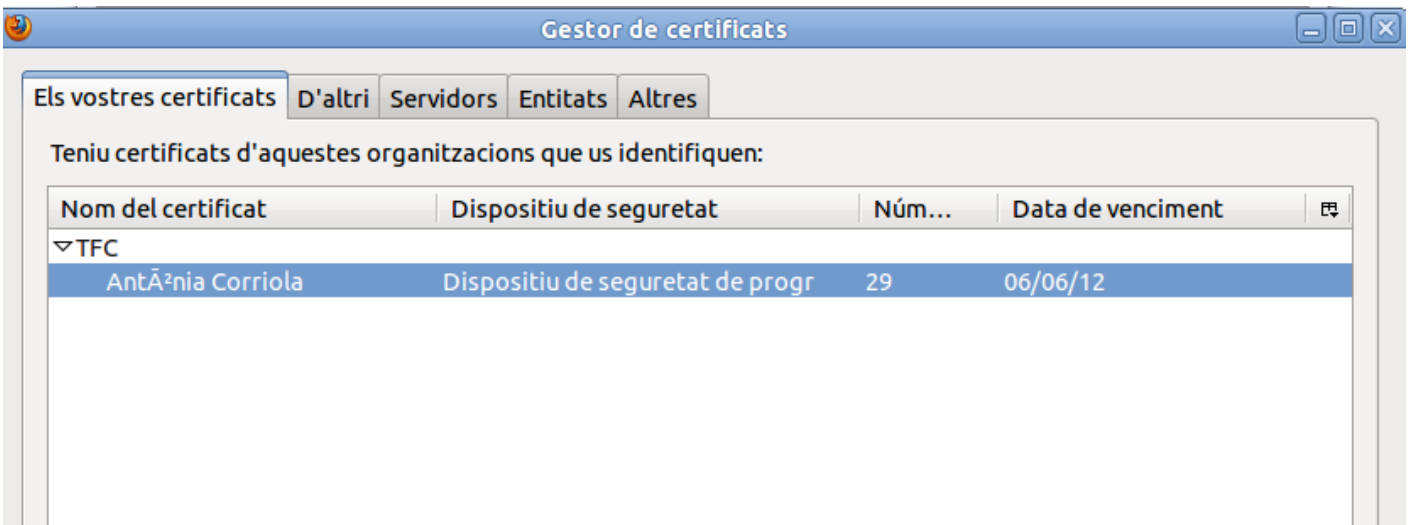


Captura 18: Importació del certificat al navegador



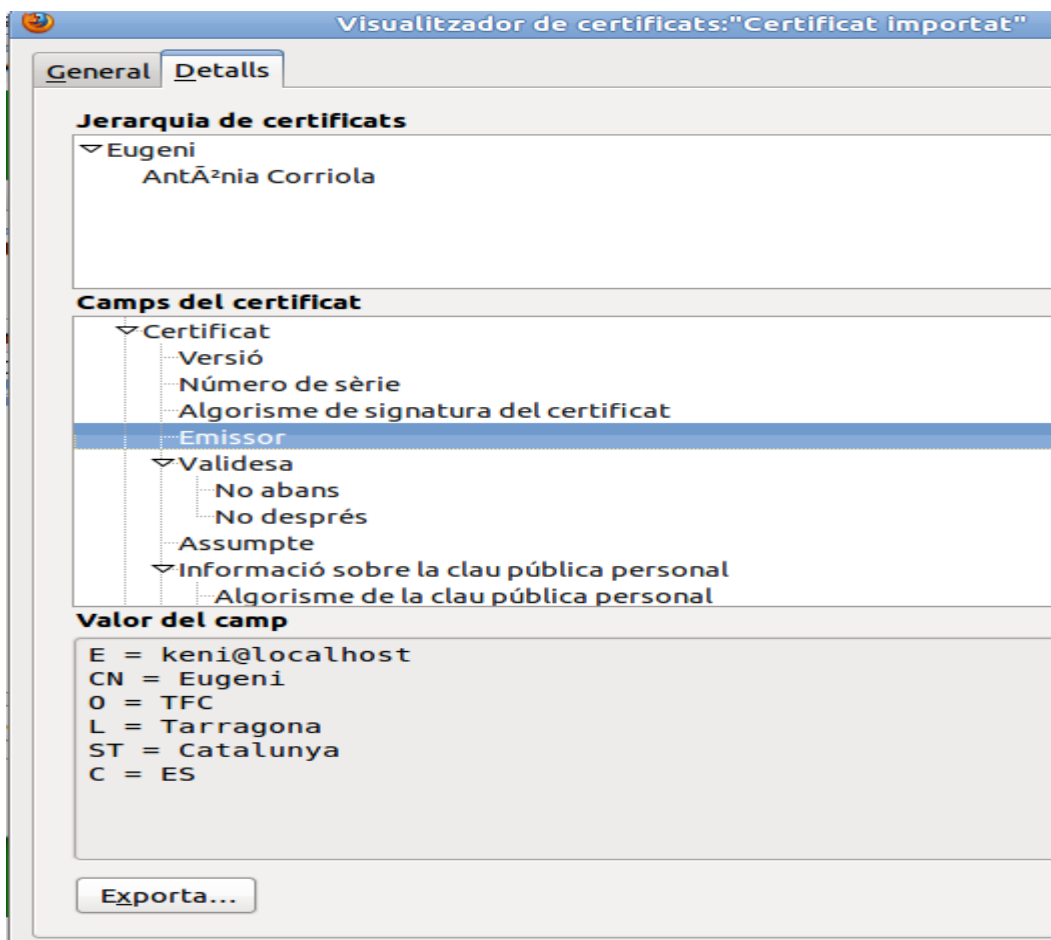
Captura 19: Ens demana la contrasenya del nostre certificat (la del client)

Després d'entrar la contrasenya, podrem veure els detalls del certificat del client:



Captura 20: Certificat PKCS#12 Importat

Detalls del certificat:



Consulta de la llista de revocació



Llista de revocació

Certificats revocats.


Lolita Clavel	Internet Widgits Pty Ltd	keni@localhost.local.lan
Perico Muntanal	Internet Widgits Pty Ltd	keni@localhost.local.lan
Ramon VxC3xA0zquez	Internet Widgits Pty Ltd	keni@localhost.local.lan
Rakel Lapiedra	VinosR	keni@localhost.local.lan
Nom complet	a	keni@localhost.local.lan

Arxiu CRL.

```

-----BEGIN X509 CRL-----
MIICFzCCAYACAQEwDQYJKoZIhvcNAQEFBQAwczELMAkGA1UEBhMCRVMEjAQBGNV
BAgTCUNhdGFsdW55YTESMBAGA1UEBxMJVGFYcmFnb25hMQwwCgYDVQQKEwNURkMx
DzANBgNVBAMTBkVlZ2VuaTEuZEdMBsGCSqGSIb3DQEJARY0a2VuaUBsb2NhbGhvc3QX
QTEwMDYwNzEyNTUwMVQoXDTEwMDYwNzEyNTUwMVQoZDQwEgIBCBcNMTEwNTAyMTAy
MjE1WjASAgEJFw0xMTA1MDIxMDIyMDIaMBICAQoXDTEwMDUwMjE1WjE1OFowEgIB
CxcNMTEwNTAyMTAyMTQ5WjASAgE0Fw0xMTA2MDMyMjE1M3MTJmBIcARMXDTEwMDYw
NjA5NDgyM1owEgIBFBcNMTEwNjA2MDk0ODI0WjASAgEWFw0xMTA2MDYwOTA1NDBa
MBICARoXDTEwMDYwNjA5NDgyNFowEgIBHBcNMTEwNjA2MDk0ODI0WqAOMAwwCgYD
VR0UBAMCAQ0wDQYJKoZIhvcNAQEFBQADgYEAdqmGXieiQ0gKStxAmuv/xAexI3h
qIqQpYsMGX6cVUiLY15YL1q7zwxLXfa6T/p5QPdHkkkDdu8+cZ5xosPLnQaafZxU
2doSpKwk70/skudDkFV+USGIhbKrm9U30ZnHWWfVx3J006EmLH05psInnOm3csF7
wcFHEDT5+j3pfAo=
-----END X509 CRL-----

```



Captura 21: Llista de revocació

Si estem autenticats amb un certificat de client al sistema podem accedir a la llista de revocació i descarregar l'arxiu.

Revocació de certificats



L'usuari pot sol·licitar la revocació del seu certificat si s'autentica correctament:

[Inici](#)[Contacte](#)[Serveis](#)

Informació de l'usuari

Informació necessària **continguda al certificat** per tramitar la sol·licitud de revocació.

Nom Complert(*)

Correu Electrònic(*)

Contrasenya(*)

Motivació per la sol·licitud de revocació.

Explicació breu

Captura 22: Sol·licitud de revocació

Cal omplir les dades demanades per que es trameti aquesta sol·licitud. Aquest servei es realitzarà com un servei "**amb certificat**" es a dir, a més d'autenticar-nos en la forma indicada fer-ho amb el nostre certificat de client, això seria més adequat. Més endavant està més ben explicat el procés.

Si tot va bé...

[Inici](#)[Contacte](#)

El certificat està present a la BD:

Ramoncin Tocamelculin
SGAE
HijosDePura
laramona@pechugona.es

S'ha tramitat la sol·licitud de revocació

L'administrador veurà després els certificats susceptibles de ser revocats:

The screenshot shows the 'Certificats no revocats' section of the TFC Safe Key CA interface. It features a green header with navigation links for 'Inici', 'Contacte', and 'Serveis'. Below the header, the text reads 'Certificats no revocats' and 'Hi ha 18 certificats no revocats. Seleccionar els que es vulguin revocar.' A table lists five certificates with checkboxes for selection. The first certificate, 'Pepa Flores', is selected. The second certificate, 'Ramoncin Tocamelculin', has a status of 'Solicitada' and a reason for non-revocation: 'Motiu: No habeis pagado mis derechos de ima'.

Checkbox	Nombre	Organització	Correu	Notes
<input checked="" type="checkbox"/>	Pepa Flores	Internet Widgits Pty Ltd	keni@localhost.local.lan	
<input type="checkbox"/>	Ramoncin Tocamelculin	SGAE	laramona@pechugona.es	Solicitada Motiu: No habeis pagado mis derechos de ima
<input type="checkbox"/>	Juan Concavo	Internet Widgits Pty Ltd	keni@localhost.local.lan	
<input type="checkbox"/>	Jean Claud Van Damme	Internet Widgits Pty Ltd	keni@localhost.local.lan	
<input type="checkbox"/>	John Malcom	Internet Widgits Pty Ltd	keni@localhost.local.lan	

Captura 23: Revocació de certificats

Seleccionarà els que vulgui revocar i ho farà.

The screenshot shows the 'Certificats Revocats' section of the TFC Safe Key CA interface. It features a green header with the text 'TFC Safe Key CA' and navigation links for 'Inici', 'Contacte', and 'Serveis'. Below the header, the text reads 'Certificats Revocats' and 'Resultats de la revocació'. A table shows the results of the revocation process, with green checkmarks indicating successful revocation for four certificates.

Resultat	Nombre	Organització	Correu	Acció
✓	Ramoncin Tocamelculin	SGAE	laramona@pechugona.es	Revocat
✓	Juan Concavo	Internet Widgits Pty Ltd	keni@localhost.local.lan	Revocat
✓	Jean Claud Van Damme	Internet Widgits Pty Ltd	keni@localhost.local.lan	Revocat
✓	John Malcom	Internet Widgits Pty Ltd	keni@localhost.local.lan	Revocat

Captura 24: Resultat de la revocació

Generació de la llista de revocació



La llista de certificats revocats s'ha generat correctament

Llista de revocació

Arxiu CRL.

```

-----BEGIN X509 CRL-----
MIICaDCCAdeCAQEWdQYJKoZIhvcNAQEFBQAwezELMAkGA1UEBhMCRVMxEjAQBgNV
BAgTCUNhdGFsdW55YTESMBAGA1UEBxMjVGFYcmFnb25hMQwwCgYDVQQKEwNURkMx
DzANBgNVBAMTBkVlZ2VuaTEEdMBsGCSqGSIb3DQEJARY0a2VuaUBsb2NhbGhvc3QX
DTEwMDYwODEwMTk1OVV0XDTEwMDYwODEwMTk1OVV0wggEYMBICAQgXDTEwMDUwMjEw
MjIxNVowEgIBCRcNMTEwNTAyMTAyMjA5WjASAgEKFw0xMTA1MDIxMDIxNThaMBIC
AQsXDTEwMDUwMjEwMjE0OVowEgIBDhcNMTEwNTAzMjIzNzEyWjASAgETfw0xMTA2
MDYwOTQ0MjNaMBICARQXDTEwMDYwNTA1NDhBaMBICARcXDTEwMDYwODA4NTUyMlowEgIBGBCN
MTEwNTAzMjIzNzEyWjASAgEaFw0xMTA2MDYwOTQ0MjRaMBICARwXDTEwMDYwNTA1
NDgYNFowEgIBJRcNMTEwNTA1NDg1NTIzWqAOMAwwCgYDVROUBAMCAQ8wDQYJKoZI
hvcNAQEFBQADgYEAZvcjhIlMVxwI87U9h0xrZhlpyWkY/Iidw/FLJf3YZrP3Faz
K4aUa1XRnxo2jFVHmSxOwqVUSLfJ3c4miicttRkcx/xDLBVleiwgf58ho72uZbvz
tfe5W9oQ3+4qD0JZ17qW3hI94wdrk4g8M00apbQsMjMpIaEofzfZA7uRHns=
-----END X509 CRL-----

```

Captura 25: Llista de revocació generada

L'administrador pot generar la llista de revocació, senzillament seleccionant aquesta opció. Es podria haver automatitzat el procés, per que es faci de manera automàtica, d'una forma bastant senzilla, o bé just després del procés de revocació de forma immediata. De tota manera, així es fa més evident a l'hora de fer el seguiment de les funcionalitats.

Introducció manual de la petició en format PKCS#10



Fins aquí el sistema generava a partir de les dades subministrades pel client la clau privada del client i la petició de signatura de forma automàtica. Parlant amb el consultor em va suggerir la possibilitat de prescindir de generar (i menys de desar) la clau privada del client a la pròpia CA, sinó permetre que el client generi la seva pròpia clau i petició de signatura i subministri aquesta última a la CA per que la signi i obtingui el corresponent certificat.

Tot això mantenint la clau privada del client en el seu poder. Això també permet que el client generi la seva clau xifrada amb l'algorisme i el nombre de bits que consideri oportuns. Cosa que en principi és desitjable.

Per tant, com a clients de la CA, generarem una clau privada i una petició de certificació:

```
~$ openssl genrsa -out key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
~$
```

```
~$ openssl req -new -key key.pem -out req.pem
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]:Jaen
Locality Name (eg, city) []:Jaen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GispyCorp Ltd
Organizational Unit Name (eg, section) []:VinijosPatoos
Common Name (eg, YOUR name) []:Joze ErGitano
Email Address []:keni@localhost.local.lan

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

Ara obriríem la petició amb el editor de text per copiar el contingut dins el quadre de text que proporciona l'aplicatiu. L'aplicació extraurà de la petició les dades que necessiti per realitzar la signatura d'acord amb la política de la CA i si cal demanarà que es generi un nova petició.

Procedirem doncs tal com em dit:



Petició de l'usuari

Si heu generat la vostra pròpia petició pkcs#10, Enganxeu tot el contingut des de -----BEGIN CERTIFICATE REQUEST----- Fins -----END CERTIFICATE REQUEST----- , inclosos al següent quadre de text:

PKCS#10(*)

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC4jCCAcOCAQAwwZwx CzAJBgNVBAYTAkVTMQ0wCwYDVQQIEwRKYWVudM00wCwYD
VQQHEwRKYWVudMRYwFAYDVQQKEw1HaXNweUNvbnAgTHRkMRYwFAYDVQQLEw1WaW5p
am9zUGF0b29zMRywFAYDVQQDEw1Kb3plIEVyb2l0YW5vMScwJQYJKoZIhvcNAQkB
Fhh rZW5pQGxvY2FsaG9zdC5sb2NhbC5sYW4wgG EiMA0GCsqGSIB3DQEBAQUAA4IB
DwAwggEKAoIBAQC L2j1JM s02WVXkD6UsX3e8k00nB5kZjGrC1RyaCUakZrgxFn63
k3c2mYKI dCQSuH2VXd iHWv/1CV5fQt f2YHV4vywDmYY3s3Dx6DvtCwr2Dgdze4bo
kCvyw34YTeckuggdRoqccWl7Zmnh3u1ixE4Su5Xj cTaEUSUf7eBqqI+QSTp+z1Pk
VxH+Sg1qoQfSqHMeYl4rfkmt r6j d3jBBG9l1pQqpF3gAw29sxQImYw5J9xuaCFAi
```

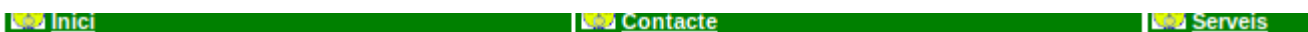
Caldrà introduir també una contrasenya per registrar l'usuari a la base de dades.

Contrasenya(*)

Nota: La petició ha d'incloure una adreça de correu electrònic vàlida.

Captura 26: Introducció de PKCS#10 directament

Si tot va bé ja tenim la petició formulada, se'ns demana confirmació, i ja podem signar la petició:



Les dades ha estat ben introduïdes:

subject=/C=ES/ST=Jaen/L=Jaen/O=GispyCorp Ltd/OU=VinijosPatoos/CN=Joze ErGitano/emailAddress=keni@localhost.lc

Joze ErGitano
GispyCorp Ltd
SPAIN
Jaen
Jaen
VinijosPatoos
keni@localhost.local.lan

S'ha enviat un codi secret al correu proporcionat, introduiu-lo al següent quadre de text.

Validació per Email

Introduiu el text xifrat aquí:

Secret

El moment de generar el certificat PKCS#12 se'n demanarà la clau privada, per generar-lo i emetre'l però no serà emmagatzemada en cap cas. De tota manera, si no es vol emetre aquest certificat, el tenim en format PEM per copiar-lo pel nostre compte.

El certificat està present a la BD:

Jose ErGitano
GispyCorp Ltd
VinijosPatoos
keni@localhost.local.lan

Emissió de certificat

- Si es vol emetre el certificat en format PKCS#12, cal enganxar (si no ho està) al següent quadre de text La clau privada xifrada amb la contrasenya proporcionada.
- Si la clau privada es troba ja a la BD i es vol eliminar després d'emetre el certificat seleccionar el quadre corresponent.

Certificat PEM:

```
-----BEGIN CERTIFICATE-----
MIIDHTCCAUGAwIBAgIBKjANBgkqhkiG9w0BAQUFADBzMQswCQYDVQGEwJFuzES
HBAEAlUECBJ0ZFOYwziLnLHMRIwEAYDVQHEwLUDYXJyYWRvdjEwEzA0MjE0
A1R60zEPHABGAlUEA+NGRXVnZw5pRR0vGwY3KozIhvcNAQ0BFg5vZm5p06+vY2Fs
aG9zd0AeFw0zMTAzMDg0MTA5MjhaFw0zMTAzMDc0MTA5MjhaMIICGAMQswCQYDVQGE
EwJFuzENMAAGAlUECBMESwFlbjENMAAGAlUEB+MESwFlbjENMB0GAlUECHMNRZLz
cHlob3JvIEw0ZDZEMHBOGAlUEC+MNVmLUAwMjE0MjE0MjE0MjE0MjE0MjE0MjE0MjE0
ZSBFckdpdGFubzEnMCUGCSqGSIb3D0EJARYYa2VumUBab2NhbGhvc30ubG9jYVww
bGFuMIIBIjANBgkqhkiG9w0BAQEAFAAOCAQAMIBCGKCAQEApo95TLdLlV5A+L
LF92vJKNJweZGYzqvtUcmglGpGa4MRZ+tSN3NpmCiHQkErh9LV3Yh1r/90Lx0LX
9mB1eL8sA5m6N7Wv8eg1b0sk9g4Hc3uG6JAr8sN+GE3nJLoIHUaKnHFpe2Zp4d7t
YaR0EruV43E2hFELH+3gagPjkk6fa5T5FcR/koNaqEH0qhzHndek35Jr+e+o3d4w
QRvZdaUkqRd4ANNvbMUCjM05Fcbnggh0IqV90v7MD4IZMmrslYH4ET7h0zEZZMuf
euHrw13juVkaY0SH9AxMr7r35aXFZ/oyBqLLmIsmu6E8FLD1EmFj0ywZnxyp50
RIwPz0IDAQABo3swETAJBgNVHRMEAjAAACwGCMGSAAG+EI8000FFH1PcGVuU1NM
IEedLbmYyXRRLZCB0ZDJ0emZpY2F0ZTAdBgNVHQ4EFgQU5D6eA0pvEYCOUzIsmmD3
-----
```

Clau privada xifrada:

Esborrar clau privada de la BD

Generar certificat PKCS#12

Si enganxem la clau privada, en generarà el corresponent certificat PKCS#12, igual que abans.

Autenticació de les funcions administratives a través del certificat digital expedit per la nostra CA



Fins ara el sistema d'autenticació de l'administrador és una contrasenya que permet accedir a les diferents funcions administratives a través de la capa de sessions de PHP. Aquest esquema de seguretat pot resultar feble si es té en compte aquesta contrasenya es troba d'alguna manera (encara que sigui xifrada) en el codi font, per tant descobrir aquesta contrasenya deixa tot el sistema obert a qualsevol.

Per evitar aquesta circumstància, podem generar un certificat per l'administració del sistema i no permetre l'accés (via ssl) al directori admin/ del lloc si no es verifica el certificat de l'administrador. Afegit a això es mantindrà la capa de seguretat que ja teníem implementada amb el PHP (sessions).

Seguirem els següents passos:

1. Podem generar un certificat vàlid per a un usuari anomenat **Administrador TFC** tal com havíem fet abans (o amb línia de comandes via openssl).
2. Amb això obtenim el certificat PKCS#12. Ometo el passos ja que ja estan descrits anteriorment.
3. Els detalls del certificat:

S'ha verificat el certificat per als usos següents:

Certificat de client SSL

Certificat de servidor SSL

Certificat de signatura de correu electrònic

Certificat de destinatari de correu electrònic

Emès a nom de

Nom habitual (CN)	Administrador TFC
Organització (O)	ftcCA
Unitat organitzacional (OU)	Administrador
Número de sèrie	30

Emès per

Nom habitual (CN)	Eugeni
Organització (O)	TFC
Unitat organitzacional (OU)	<No forma part del certificat>

Validesa

Data d'emissió	10/06/11
Data de venciment	09/06/12

Empremtes digitals

4. Ara editarem l'arxiu **/etc/apache2/sites-available/default-ssl** per que quedi com segueix (S'han afegit comentaris en negreta):

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
  ServerAdmin webmaster@localhost

  DocumentRoot /var/www/
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  # Aquí el directori dels usuaris
  # No hi ha (de moment) cap directori amb autenticació per certificat
  <Directory /var/www/tfc/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>

  # Aquí afegirem regles de control d'accés
  # a l'àrea d'administració
  <Directory /var/www/tfc/admin/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
    SSLVerifyClient require
    SSLRequire (%{SSL_CLIENT_S_DN_CN} eq "Administrador TFC")
  </Directory>
```

Com es pot veure només demanarem autenticació SSL per tenir accés al directori **/var/www/tfc/admin/** que és on tenim definides les funcions administratives de la CA. A més es verificarà el nom del propietari (es poden afegir més comprovacions en sintaxi PERL) del certificat.

També modificarem les línies següents, que han de quedar com a sota:

```
# Enable/Disable SSL for this virtual host.
SSLEngine on
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/cacert.pem
SSLCertificateKeyFile /etc/ssl/private/cakey.pem
# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCACertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
SSLCACertificatePath /etc/ssl/newcerts/
SSLCACertificateFile /etc/ssl/certs/cacert.pem
# ...
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
SSLCARevocationPath /etc/ssl/crl/
SSLCARevocationFile /etc/ssl/crl/revocats.crl
```

5. Un cop fet això reiniciem el servidor amb la nova configuració (demana la contrasenya de la CA)

```
# /etc/init.d/apache2 restart
* Restarting web server apache2
... waiting Apache/2.2.16 mod_ssl/2.2.16 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

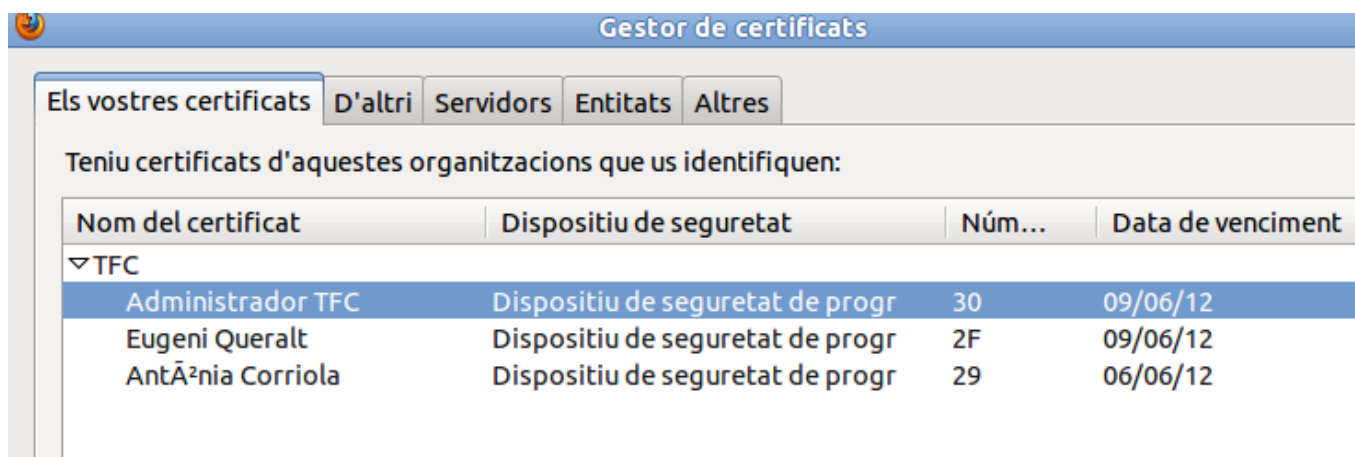
Server localhost.localdomain:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.

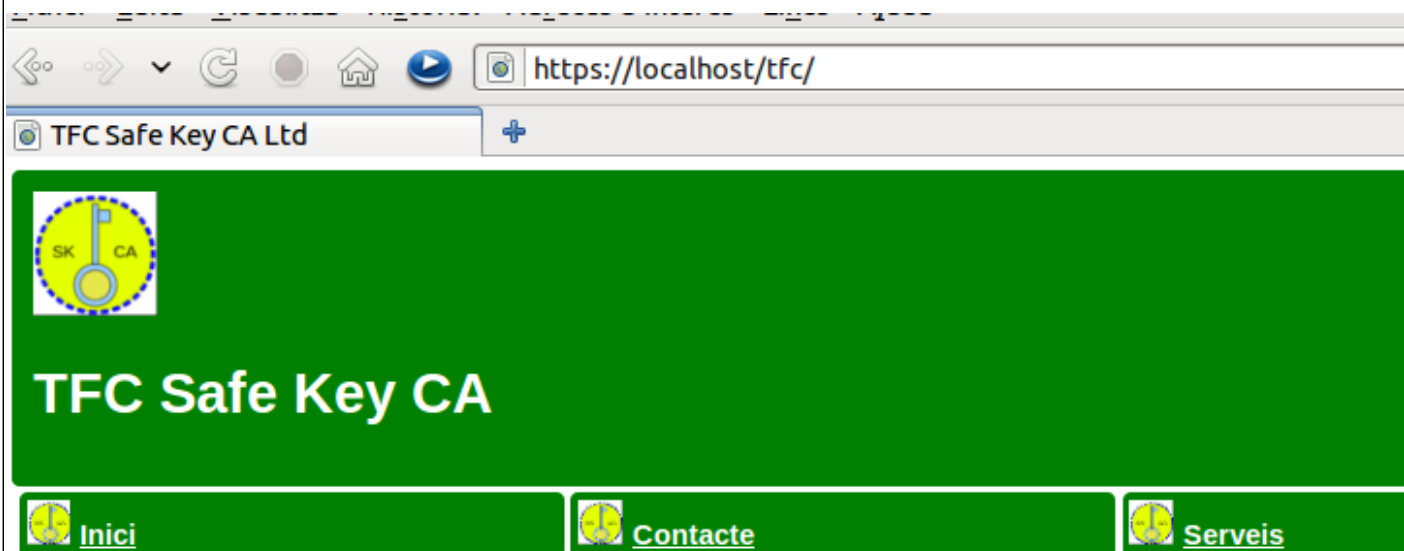
OK ]
#
```

[

6. Importem el nostre certificat al navegador tal com s'ha explicat abans i queda de la següent manera:

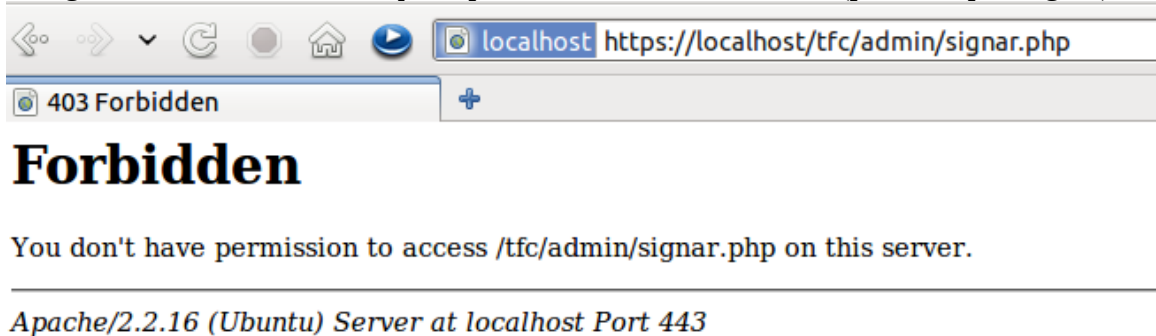


7. Anem ara al nostre lloc segur:

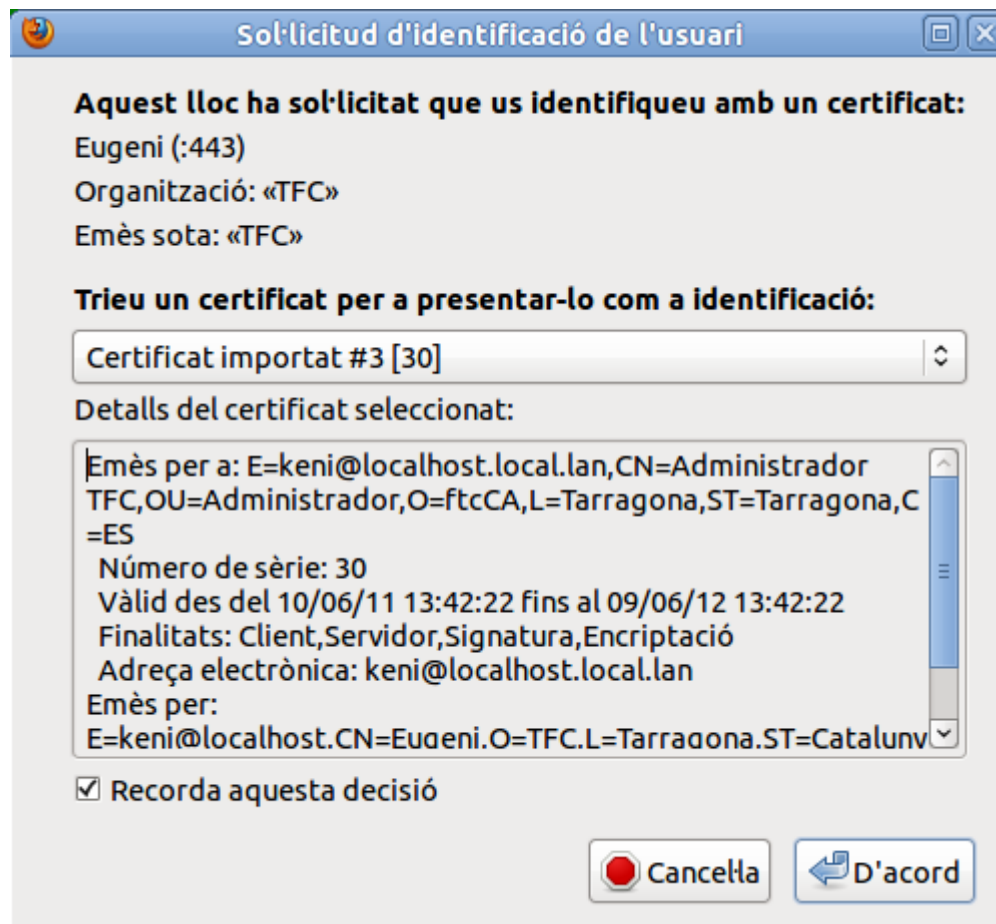


I ara si intentem accedir a les funcions administratives se'ns demanarà que ens autentifiquem amb el nostre certificat de client, que si no es correspon amb el de l'administrador (segons les regles d'accés definides a l'arxiu /etc/apache2/sites-available/default-ssl, que el nostre cas només és el CN del client) ens generarà un missatge d'error. Com segueix...

8. Missatge d'error si no en autèntiquem per funcions administratives (per exemple, signar):



9. Si seleccionem el certificat adequat (d'administrador) ja podrem entrar a fer tasques administratives, **no ens hem d'oblidar de posar la contrasenya d'administrador** per iniciar la sessió de PHP.




Captura 27: Ens demana el certificat de client de l'usuari administrador


Amb això ja estem autènticats amb el nostre certificat d'administrador i dins la sessió administrativa de la CA.

Autenticació dels clients amb certificat

Hi ha serveis descrits abans com són; l'accés a la llista de certificats revocats (es podria optar per fer-lo públic) com la sol·licitud de revocació de certificats que poden ser accedit a través d'autenticació amb certificat, és per això que he creat una nova carpeta, anomenada **ambCertificat**/ dins l'arbre de la web i hi he traslladat aquests 2 serveis a dins. He modificat l'aspecte de la pàgina principal, que ara apareixerà com a la captura següent:



CA SK, és una autoritat de certificació creada per donar una sèrie de serveis de signatura de certificats i autenticació en línia. Tant per les tasques administratives (**Administrador**) com per als **Serveis amb certificat** es demana una autenticació a través del certificat de client corresponent.

TFC Safe Key CA Ltd.

 This work is licensed under a [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

Després, igual que a l'anterior apartat he modificat l'arxiu `/etc/apache2/sites-available/default-ssl` per que llueixi així:

```
# Aquí permetem l'accés al serveis públics
</Directory>
<Directory /var/www/tfc/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>

# Aquí demanem autenticació
# a l'usuari amb certificat
<Directory /var/www/tfc/ambCertificat>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
    SSLVerifyClient require
</Directory>
```

Reiniciariem el servidor apache2, i ja tenim controlat l'accés als serveis amb certificat emès per la nostra CA.

Estructura de la Base de Dades



```
-- phpMyAdmin SQL Dump
-- version 3.3.7deb5build0.10.10.1
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Temps de generació: 07-06-2011 a les 15:01:58
-- Versió del servidor: 5.1.49
-- Versió de PHP : 5.3.3-1ubuntu9.5

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de dades: `tfcCA`
--
-----

--
-- Estructura de la taula `certificats_clients`
--

CREATE TABLE IF NOT EXISTS `certificats_clients` (
  `crt_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `crt_descripc` varchar(128) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `data_creacio` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `passwd` text CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `commonName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `organizationName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `organizationalUnitName` text CHARACTER SET utf8 COLLATE utf8_swedish_ci,
  `emailAddress` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `countryName` char(2) CHARACTER SET utf8 COLLATE utf8_spanish_ci DEFAULT NULL,
  `stateOrProvinceName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `localityName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `revocat` tinyint(1) NOT NULL DEFAULT '0',
  `solrev` tinyint(1) NOT NULL DEFAULT '0',
  `motiurev` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `certpem` text CHARACTER SET utf8 COLLATE utf8_unicode_ci,
  PRIMARY KEY (`crt_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=26 ;

-----
```

```
--
-- Estructura de la taula `peticions_csr`
--

CREATE TABLE IF NOT EXISTS `peticions_csr` (
  `csr_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `csr_descripc` text CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `date_creacio` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `passwd` text CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `commonName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `organizationName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `organizationalUnitName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `emailAddress` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `countryName` char(2) CHARACTER SET utf8 COLLATE utf8_spanish_ci DEFAULT NULL,
  `stateOrProvinceName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `localityName` text CHARACTER SET utf8 COLLATE utf8_spanish_ci,
  `signada` tinyint(1) NOT NULL DEFAULT '0',
  `pkcs10` text COLLATE utf8_unicode_ci NOT NULL,
  `pkeypem` text CHARACTER SET ucs2 COLLATE ucs2_unicode_ci,
  PRIMARY KEY (`csr_id`),
  KEY `csr_id` (`csr_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=39 ;

-----

--
-- Estructura de la taula `usuaris`
--

CREATE TABLE IF NOT EXISTS `usuaris` (
  `user_id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` text COLLATE utf8_spanish_ci NOT NULL,
  `user` text COLLATE utf8_spanish_ci NOT NULL,
  `email` text COLLATE utf8_spanish_ci NOT NULL,
  `passwd` text COLLATE utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci AUTO_INCREMENT=4 ;
```

La base de dades és molt senzilla, les 2 primeres taules emmagatzemen les dades de les peticions i certificats expedits, la d'usuaris és per l'autenticació dels mateixos al sistema. Les contrasenyes es desen xifrades per evitar el compromís de les dades en cas d'un accés no autoritzat. Respecte la clau privada dels clients es desa juntament amb les peticions de forma **temporal**, un cop emès el certificat del client aquesta s'esborra a petició del client.

Implementació i estructura del programari.

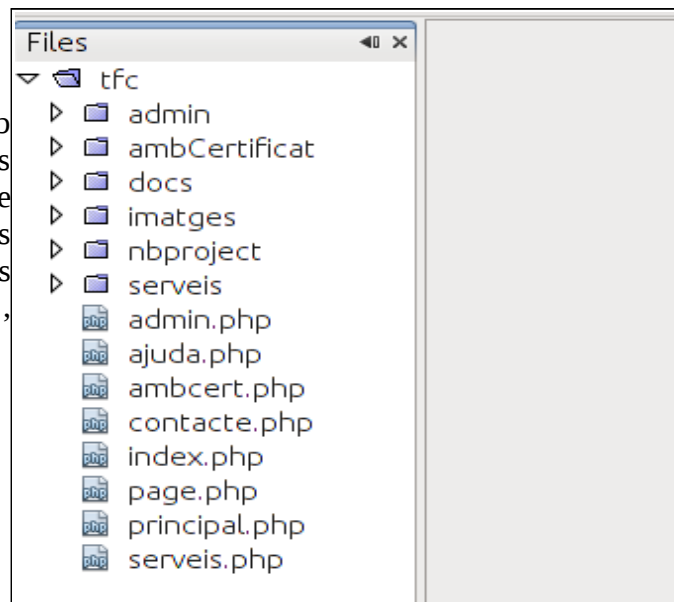


El codi font es pot trobar al directori tfc/ està creat de forma íntegra en llenguatge PHP i present una estructura modula que s'aprofita de la POO per mantenir un aspecte uniforme mentre s'executa l'aplicació. Tota l'aplicació corre sota HTTPS d'aquesta manera assegurem el xifratge asimètric (amb la clau i certificat de la nostra CA) de les comunicacions, ja que son molts els formularis i dades sensibles transferits.

A més incorporem una potent característica de seguretat en PHP (no li és exclusiu), el control de sessions, això afegeix una capa més de seguretat a la nostra aplicació, que permetrà l'autenticació, tant d'usuaris com de l'administrador per accedir a les diferents funcions i serveis.

La estructura modular fa que sigui relativament senzill afegir millores i serveis addicionals a l'entorn, tot i que es pot millorar molt la tetrarquia de les classes i la encapsulació d'alguns mètodes com per exemple afegir una capa d'abstracció de bases de dades (d'això m'he adonat al final), millorar les opcions de registre de nous usuaris, etc.

Aquí es mostra la estructura del treball (fet amb l'editor NetBeans IDE6.9). Els arxius que es mostren, corresponen al menú principal mentre que els serveis públics serveis amb certificat i les funcions administratives estan confinades als directoris *serveis*, *ambCertificat* i *admin*, respectivament.



La documentació del treball es troba al directori /docs (adjuntat fora del directori /tfc/) proporcionat dins l'arxiu comprimit del lliurament. S'ha generat en format HTML amb un anàleg del javadoc anomenat **PhpDocumentor**. Per geremar la documentació s'ha fet :

```
~$ phpdoc -o HTML:frames:earthli -d tfc/ -t docs/
```

A la carpeta /docs es poden veure els diferents comentaris que s'han realitzat sobre els mòduls i les classes.

Bibliografia



- Canonical Ltd. and members of the Ubuntu Documentation Project :
“Ubuntu Server Guide” Copyright © 2010
- Chris Snyder, Thomas Myer, and Michael Southwell Copyright © 2010 : **“Pro PHP Security From Application Security Principles to the Implementation of XSS Defenses”** . Second Edition . Apress ®
- Helena Rifà Pous : **“Infraestructura de clau pública PKI ”** Tercera edició: febrer 2006 © Universitat Oberta de Catalunya
- Luke Welling, Laura Thompson: **“Desarrollo Web con PHP i MySQL”**. 3a. Ed. Ediciones Anaya Multimedia, © 2005
- Matt Doyle: **“Fundamentos PHP Práctico ”** Ediciones Anaya Multimedia © 2010

Webgrafia



- <http://bulma.net/body.phtml?nIdNoticia=2285>
- <http://cryptophp.wikidot.com/ejemplo-openssl-pkcs12-export>
- <http://helektron.com/tutorial-como-crear-una-autoridad-certificadora-ca-con-openssl/>
- <http://manual.phpdoc.org>
- <http://netbeans.org/kb/trails/php.html>
- <http://php.net/>
- <http://www.apache.org/>
- http://www.bdat.net/documentos/certificados_digitaes/
- <http://www.desarrolloweb.com/manuales/37/>
- <http://www.forosdelweb.com/wiki/PHP>
- <http://www.linuxtotal.com.mx/>
- <http://www.madboa.com/geek/openssl>
- <http://www.maestrosdelweb.com/editorial/phpreusr/>
- <http://www.openssl.org/>
- <http://www.todoexpertos.com/categorias/tecnologia-e-internet/desarrollo-de-sitios-web/php>
- <https://help.ubuntu.com/10.04/serverguide/C/index.html>