INFORME LABORATORIO

Práctica 10: Creación de una Autoridad Certificadora (CA) con OpenSSL

Rafael Herrero Álvarez. Grupo L1.
Seguridad en Sistemas Informáticos. Turno de mañana

Contenido

1. Primera parte de la práctica

1.1. Generación de la Autoridad Certificadora

Lo primero que haremos será la instalación de una autoridad certificadora, OpenSSL, que en nuestro caso ya se encontraba instalada, utilizando el siguiente comando. Como claves y frases de paso, utilice en todo momento daniel1234:

```
rafa@rafa:~$ sudo su
[sudo] password for rafa:
root@rafa:/home/rafa# apt install openssl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssl ya está en su versión más reciente (1.0.2g-
1ubuntu4.1).
fijado openssl como instalado manualmente.
```

Tras esto nos movemos hasta el directorio en el que trabajaremos y crearemos la autoridad certificadora:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl req -
x509 -newkey rsa:2048 -days 1095 -keyout CAkey.pem -out
CAcert.pem
Generating a 2048 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]:Tenerife
Locality Name (eg, city) []:La Laguna
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:ULL
Organizational Unit Name (eg, section) []:ETSII
Common Name (e.g. server FQDN or YOUR name) []:ull.es
Email Address []:rafael.herrero.13@ull.edu.es
```

1.2. Generación del certificado del servidor

Ahora nos ocuparemos de los certificados por el lado del servidor. Los tres primeros comandos son para la configuración de los certificados y el último para emitirlo.

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl genrsa
-des3 -out serv-priv.pem -passout pass:daniel1234 2048
Generating RSA private key, 2048 bit long modulus
.......................................+++
.....................+++
e is 65537 (0x10001)

rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl req -
new -subj "/DC=root.com/OU=com/CN=root" -key serv-priv.pem
-passin pass:daniel1234 -out petic-certificado-serv.pem
```

Creamos un archivo “config1.txt” con los siguientes atributos de configuración:

```
basicConstraints = critical,CA:FALSE
extendedKeyUsage = serverAuth

rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl x509 -
CA CAcert.pem -CAkey CAkey.pem -req -in petic-certificado-
serv.pem -days 15 -extfile config1.txt -sha1 -
CAcreateserial -out servidor-cert.pem
Signature ok
subject=/DC=root.com/OU=com/CN=root
Getting CA Private Key
Enter pass phrase for CAkey.pem:
```

Los archivos generados hasta el momento son los que siguen:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ ll
total 36
drwxrwxr-x  2 rafa rafa 4096 may 10 10:16 ./
drwxrwxr-x 12 rafa rafa 4096 may 10 10:05 ../
-rw-rw-r--  1 rafa rafa 1432 may 10 10:11 CAcert.pem
-rw-rw-r--  1 rafa rafa   17 may 10 10:16 CAcert.srl
-rw-rw-r--  1 rafa rafa 1834 may 10 10:11 CAkey.pem
-rw-rw-r--  1 rafa rafa   67 may 10 10:15 config1.txt
-rw-rw-r--  1 rafa rafa  940 may 10 10:14 petic-
certificado-serv.pem
-rw-rw-r--  1 rafa rafa 1253 may 10 10:16 servidor-cert.pem
-rw-rw-r--  1 rafa rafa 1751 may 10 10:13 serv-priv.pem
```

1.3. Generación de los certificados de los clientes

Hasta ahora hemos generado los certificados de parte del servidor, por lo que en este momento generaremos los del cliente. Ejecutaremos los siguientes comandos:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl genrsa
-des3 -passout pass:daniel1234 -out client-priv.pem 2048
Generating RSA private key, 2048 bit long modulus
.................................+++
.....................................................
.........+++
e is 65537 (0x10001)

rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl req -
new -key client-priv.pem -passin pass:daniel1234 -subj
"/DC=localhost/OU=com/CN=Fsv" -out petic-cert-client.pem
```

Creamos un archivo “config2.txt” con los siguientes atributos de configuración:

```
basicConstraints = critical,CA:FALSE
extendedKeyUsage = clientAuth

rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl x509 -
CA CAcert.pem -CAkey CAkey.pem -req -in petic-cert-
client.pem -set_serial 3 -days 15 -out client-cert.pem -
extfile config2.txt
Signature ok
subject=/DC=localhost/OU=com/CN=Fsv
Getting CA Private Key
Enter pass phrase for CAkey.pem:
```

Los archivos generados hasta el momento son los que siguen:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ ll
total 48
drwxrwxr-x  2 rafa rafa 4096 may 10 10:19 ./
drwxrwxr-x 12 rafa rafa 4096 may 10 10:05 ../
-rw-rw-r--  1 rafa rafa 1432 may 10 10:11 CAcert.pem
-rw-rw-r--  1 rafa rafa   17 may 10 10:16 CAcert.srl
-rw-rw-r--  1 rafa rafa 1834 may 10 10:11 CAkey.pem
-rw-rw-r--  1 rafa rafa 1241 may 10 10:26 client-cert.pem
-rw-rw-r--  1 rafa rafa 1743 may 10 10:18 client-priv.pem
-rw-rw-r--  1 rafa rafa   67 may 10 10:15 config1.txt
-rw-rw-r--  1 rafa rafa   67 may 10 10:19 config2.txt
-rw-rw-r--  1 rafa rafa  940 may 10 10:18 petic-cert-
client.pem
-rw-rw-r--  1 rafa rafa  940 may 10 10:14 petic-
certificado-serv.pem
-rw-rw-r--  1 rafa rafa 1253 may 10 10:16 servidor-cert.pem
-rw-rw-r--  1 rafa rafa 1751 may 10 10:13 serv-priv.pem
```

1.4. Exportando los certificados de los clientes

Una vez obtenidos todos los certificados, debemos de exportar los mismos para que se puedan utilizar en programas externos como un navegador o un gestor de correo. Ejecutamos el siguiente comando para obtener un fichero en formato p12:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl pkcs12
-export -in client-cert.pem -inkey client-priv.pem -
certfile CAcert.pem -out cert-pck12.p12
Enter pass phrase for client-priv.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

1.5. Definiendo la lista de revocación

Además de poder generarlos, también deberíamos poder revocarlos en caso de que no nos interesase que un certificado se pudiese seguir utilizando. Para ello, debemos de tener una lista con los que no nos interese. En nuestro caso trabajamos sobre el mismo directorio anterior. Ejecutamos los siguientes comandos, primero creando el fichero index.txt y luego modificando el fichero de configuración de OpenSSL:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ touch
index.txt

rafa@rafa:~/Documentos/Repositorios/SSI/P10$ cd
/usr/lib/ssl/
rafa@rafa:/usr/lib/ssl$ sudo nano openssl.cnf
```

Este sería el resultado de la parte CA_default en el fichero openssl.cnf:

```
#########################################################
#########
[ CA_default ]

dir          = /home/rafa/Documentos/Repositorios/SSI/P10
certs        = $dir/certs
crl_dir      = $dir/crl
database     = $dir/index.txt
#unique_subject   = no
new_certs_dir     = $dir/newcerts
certificate  = $dir/CAcert.pem
serial       = $dir/serial
#crlnumber   = $dir/crlnumber
#crl         = $dir/crl.pem
private_key  = $dir/CAkey.pem
RANDFILE     = $dir/.rand
x509_extensions   = usr_cert
```

Ahora creamos la lista y revocamos el certificado que hemos generado anteriormente con los siguientes comandos:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl ca -
gencrl -out listarev.crl
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for
/home/rafa/Documentos/Repositorios/SSI/P10/CAkey.pem:

rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl ca -
gencrl -out listarev.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for
/home/rafa/Documentos/Repositorios/SSI/P10/CAkey.pem:
```

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl ca -
revoke client-cert.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for
/home/rafa/Documentos/Repositorios/SSI/P10/CAkey.pem:
Adding Entry with serial number 03 to DB for
/DC=localhost/OU=com/CN=Fsv
Revoking Certificate 03.
Data Base Updated

rafa@rafa:~/Documentos/Repositorios/SSI/P10$ openssl ca -
gencrl -out listarev.crl
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for
/home/rafa/Documentos/Repositorios/SSI/P10/CAkey.pem:
```

2. Segunda parte de la práctica

En esta parte extraeremos la información de un certificado que tengamos almacenado en nuestro ordenador. En nuestro caso utilizaremos el certificado generado en la primera parte de la práctica, por lo que no realizamos el apartado 2 del PDF de la práctica.

2.1. Extrayendo información de un certificado con OpenSSL

Primero convertimos el fichero p12 a pem, mostramos la calve pública por consola y luego extraemos la clave pública y privada con los siguientes comandos:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10/SegundaParte$
openssl pkcs12 -in cert-pck12.p12 -out cert-pck12.pem -
clcerts
Enter Import Password:
MAC verified OK
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

rafa@rafa:~/Documentos/Repositorios/SSI/P10/SegundaParte$
openssl x509 -text -in cert-pck12.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 3 (0x3)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Tenerife, L=La Laguna, O=ULL,
OU=ETSII,
CN=ull.es/emailAddress=rafael.herrero.13@ull.edu.es
        Validity
            Not Before: May 10 09:26:10 2016 GMT
            Not After : May 25 09:26:10 2016 GMT
        Subject: DC=localhost, OU=com, CN=Fsv
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:

00:c1:52:58:78:fb:1a:92:1b:6d:ee:5b:df:0f:fc:
cb:9a:a9:39:e1:53:df:6d:9a:54:2f:ea:2c:12:39:
b8:01:05:84:5a:80:1e:be:f0:16:e5:fd:50:82:37:
ca:23:8a:fb:0a:3d:f1:12:58:8e:27:4f:13:d9:ed:
52:4c:ef:88:32:a9:63:36:ce:18:36:83:25:94:b8:
4f:9d:74:45:83:c8:8c:56:cd:a1:35:69:99:a1:21:
2c:14:62:09:97:b6:d7:03:d7:0f:6a:ed:7c:82:0e:
fb:f0:2c:a6:af:76:fd:54:f8:ac:bc:46:1c:9a:51:
c9:55:a7:af:ae:fb:bc:3f:9f:99:9e:d8:0b:02:02:
76:eb:7e:50:d4:9b:b6:6c:5c:e0:aa:31:8b:62:34:
d8:11:a5:9d:f3:a0:ff:04:e9:41:2f:9a:45:e2:99:
3f:b5:af:66:27:48:a4:d8:9a:f3:c5:6a:16:97:cb:
```

```
da:c4:69:67:7d:80:3a:50:ef:2d:18:2c:fa:ae:32:
00:24:2d:ea:1d:2e:ce:fa:1b:4a:f3:64:71:c6:40:
ca:c4:72:a3:4d:13:12:05:88:05:22:5c:3e:bc:60:
5f:fe:92:df:db:3c:84:ef:06:b2:98:1e:fc:3e:9f:
5a:09:10:be:75:62:21:37:68:a2:68:e1:11:c3:7c:
62:09
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
    Signature Algorithm: sha256WithRSAEncryption

47:dd:80:ec:9a:de:fc:cc:90:78:47:28:83:a9:61:20:34:1b:
57:cb:a9:38:3d:14:e7:18:3b:f3:a0:b8:9b:a2:65:21:80:5f:
e7:82:d0:22:67:fc:4c:07:ab:db:e2:d7:3a:85:9f:2d:8c:83:
87:5f:16:e4:f1:86:66:3e:28:d3:8b:98:eb:89:fd:e2:70:6b:
cb:21:18:f2:7b:c4:be:13:e6:78:8c:96:ae:6a:e5:37:7e:7c:
8d:44:66:c5:c0:75:84:9d:c4:27:fe:dd:58:15:85:28:60:2a:
08:b7:14:5a:5d:48:c2:8e:df:8e:3e:d4:77:66:35:2c:b5:81:
d9:51:80:95:ae:15:18:ce:c2:58:3e:98:d3:93:43:75:48:b1:
81:3c:59:10:6a:61:85:02:9b:dc:6d:10:44:3e:66:dc:1f:93:
23:16:88:e1:c6:46:45:c5:e5:30:33:ca:b6:fc:26:cc:2f:d7:
39:24:1c:b6:75:c4:69:f1:82:e2:c7:d9:5c:8a:4f:b8:af:c8:
87:cd:8e:c3:bc:03:04:e4:f8:91:76:b9:8d:75:d8:8c:4d:c6:
5d:52:b2:cc:42:ad:2f:73:1b:4a:81:3b:04:b2:01:88:9c:89:
4d:5c:98:ce:a9:47:9e:e4:4e:0c:3f:f8:b4:90:a6:d6:eb:91:
8c:ed:6b:54
-----BEGIN CERTIFICATE-----
MIIDaDCCAlCgAwIBAgIBAzANBgkqhkiG9w0BAQsFADCBkDELMAkGA1UEBhM
CRVMxETAPBgNVBAgMCFRlbmVyaWZlMRIwEAYDVQQHDAlMYSBMYWd1bmExDD
AKBgNVBAoMA1VMTDEOMAwGA1UECwwFRVRTSUkxDzANBgNVBAMMBnVsbC5lc
zErMCkGCSqGSIb3DQEJARYccmFmYWVsLmhlcnJlcm8uMTNAdWxsLmVkdS5l
czAeFw0xNjA1MTAwOTI2MTBaFw0xNjA1MjUwOTI2MTBaMDcxGTAXBgoJkia
Jk/IsZAEZFglsb2NhbGhvc3QxDDAKBgNVBAsMA2NvbTEMMAoGA1UEAwwDRn
N2MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwVJYePsakhtt7
lvfD/zLmqk54VPfbZpUL+osEjm4AQWEWoAevvAW5f1QgjfKI4r7Cj3xEliO
J08T2e1STO+IMqljNs4YNoMllLhPnXRFg8iMVs2hNWmZoSEsFGIJl7bXA9c
Pau18gg778Cymr3b9VPisvEYcmlHJVaevrvu8P5+ZntgLAgJ2635Q1Ju2bF
zgqjGLYjTYEaWd86D/BOlBL5pF4pk/ta9mJ0ik2JrzxWoWl8vaxGlnfYA6U
O8tGCz6rjIAJC3qHS7O+htK82RxxkDKxHKjTRMSBYgFIlw+vGBf/pLf2zyE
7waymB78Pp9aCRC+dWIhN2iiaOERw3xiCQIDAQABoyUwIzAMBgNVHRMBAf8
EAjAAMBMGA1UdJQQMMAoGCCsGAQUFBwMCMA0GCSqGSIb3DQEBCwUAA4IBAQ
BH3YDsmt78zJB4RyiDqWEgNBtXy6k4PRTnGDvzoLibomUhgF/ngtAiZ/xMB
6vb4tc6hZ8tjIOHXxbk8YZmPijTi5jrif3icGvLIRjye8S+E+Z4jJauauU3
fnyNRGbFwHWEncQn/t1YFYUoYCoItxRaXUjCjt+OPtR3ZjUstYHZUYCVrhU
YzsJYPpjTk0N1SLGBPFkQamGFApvcbRBEPmbcH5MjFojhxkZFxeUwM8q2/C
bML9c5JBy2dcRp8YLix9lcik+4r8iHzY7DvAME5PiRdrmNddiMTcZdUrLMQ
q0vcxtKgTsEsgGInIlNXJjOqUee5E4MP/i0kKbW65GM7WtU
-----END CERTIFICATE-----
```

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10/SegundaParte$
openssl rsa -in cert-pck12.pem -out tuclave_publica.pem -
pubout
Enter pass phrase for cert-pck12.pem:
writing RSA key

rafa@rafa:~/Documentos/Repositorios/SSI/P10/SegundaParte$
openssl rsa -in cert-pck12.pem -des3 -out
tuclaveprivada.pem
Enter pass phrase for cert-pck12.pem:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Tras esto, firmaremos con la clave privada un fichero cualquiera y lo verificaremos con la clave pública:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10/SegundaParte$
echo "hola hola caracola" > Fichero.txt

rafa@rafa:~/Documentos/Repositorios/SSI/P10/SegundaParte$
openssl dgst -sha1 -sign tuclaveprivada.pem -out
Firmado.sig Fichero.txt
Enter pass phrase for tuclaveprivada.pem:

rafa@rafa:~/Documentos/Repositorios/SSI/P10/SegundaParte$
openssl dgst -sha1 -verify tuclave_publica.pem -signature
Firmado.sig Fichero.txt
Verified OK
```

3. Conclusiones y problemas

Para probar que el certificado generado para el cliente era válido, lo he introducido en el Google Chrome y lo ha reconocido perfectamente como se ve en la siguiente captura:

Una vez terminada la práctica he acabado con los siguientes ficheros en el directorio de trabajo:

```
rafa@rafa:~/Documentos/Repositorios/SSI/P10$ tree
.
└── Ficheros
    ├── CAcert.pem
    ├── CAcert.srl
    ├── CAkey.pem
    ├── cert-pck12.p12
    ├── client-cert.pem
    ├── client-priv.pem
    ├── config1.txt
    ├── config2.txt
    ├── index.txt
    ├── index.txt.attr
    ├── index.txt.old
    ├── listarev.crl
    ├── listarev.pem
    ├── petic-cert-client.pem
    ├── petic-certificado-serv.pem
    ├── salida terminal.txt
    ├── SegundaParte
    │   ├── cert-pck12.p12
    │   ├── cert-pck12.pem
    │   ├── Fichero.txt
    │   ├── Firmado.sig
    │   ├── tuclaveprivada.pem
    │   └── tuclave_publica.pem
    ├── servidor-cert.pem
    └── serv-priv.pem

2 directories, 24 files
```

Como se puede observar, el fichero index.txt para la revocación de los certificados se encuentra en el mismo directorio y no en /etc/ssl, por lo que para que funcionase únicamente tuve que modificar el fiero de configuración de openssl.conf. Además de esto, la ruta para ese fichero openssl.conf no era la correcta en mi caso, ya que me avisaba de que estaba utilizando la ruta /usr/lib/ssl, por lo que con un pequeño cambio ya tenía todo funcionando. De resto, no encontré mayores problemas, ya que la realice en mi ordenador personal, por lo que podía ejecutar algunos comandos como superusuario. A parte de esto, los ficheros a introducir en la verificación de la firma en la segunda parte no eran los ficheros correctos, por lo que tuve que poner los correctos de clave pública y clave privada.