EXTRA - Cifrado Asimétrico

A.D.G.

Antes de nada hay que crear un usuario no root pero con permisos de sudo:

Creo el usuario:

```
andreidani@andreidani-equipo:~$ sudo adduser andreisudouser
Adding user `andreisudouser' ...
Adding new group `andreisudouser' (1001) ...
Adding new user `andreisudouser' (1001) with group `andreisudouser' ...
Creating home directory `/home/andreisudouser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for andreisudouser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

Le meto en el grupo de sudo:

```
andreidani@andreidani-equipo:~$ sudo usermod -aG sudo andreisudouser
```

1. Instala un CA en un Linux

Inicio sesión antes con mi usuario no root creado: andreisudouser

1) Instalo Easy-RSA:

```
andreisudouser@andreidani-equipo:~$ sudo apt update
Obj:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se pueden actualizar 101 paquetes. Ejecute «apt list --upgradable» para verlos.
```

```
andreisudouser@andreidani-equipo:~$ sudo apt install easy-rsa
```

2) Creo un directorio para la infraestructura de clave pública:

```
andreisudouser@andreidani-equipo:~$ mkdir ~/easy-rsa
andreisudouser@andreidani-equipo:~$ ls
easy-rsa
andreisudouser@andreidani-equipo:~$ pwd
/home/andreisudouser
```

3) Creo los enlaces simbólicos:

```
andreisudouser@andreidani-equipo:~$ ln -s /usr/share/easy-rsa/* ~/easy-rsa/
```

4) Restrinjo el acceso a otras personas, dejando acceder sólo a andreisudouser:

```
andreisudouser@andreidani-equipo:~$ chmod 700 easy-rsa/
```

5) Inicio la PKI dentro del directorio easy-rsa:

```
andreisudouser@andreidani-equipo:~$ cd ~/easy-rsa
andreisudouser@andreidani-equipo:~/easy-rsa$ ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/andreisudouser/easy-rsa/pki
```

6) Creo el archivo vars para poder crear la clave privada y el certificado de su entidad de certificación:

```
andreisudouser@andreidani-equipo:~/easy-rsa$ nano vars
```

```
  GNU nano 6.2                                  vars *
~/easy-rsa/vars
set_var EASYRSA_REQ_COUNTRY     "ES"
set_var EASYRSA_REQ_PROVINCE    "Madrid"
set_var EASYRSA_REQ_CITY        "San Fernando de Henares"
set_var EASYRSA_REQ_ORG         "OrganizacionAndrei"
set_var EASYRSA_REQ_EMAIL       "andreilisanfer@gmail.com"
set_var EASYRSA_REQ_OU          "Community"
set_var EASYRSA_ALGO            "ec"
set_var EASYRSA_DIGEST          "sha512"
```

7) Para crear el certificado root público y el par de claves privadas para la CA ejecuto el comando siguiente. Pide ingresar una contraseña que siempre usaré cuando quiero interactuar con mi CA. Luego pide un nombre común, con dar a Enter me asigna automáticamente el predeterminado:

```
andreisudouser@andreidani-equipo:~/easy-rsa$ ./easyrsa build-ca
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
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.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/andreisudouser/easy-rsa/pki/ca.crt
```

Si hubiese querido no tener que introducir la contraseña cada vez que interactúe con mi CA debería de haber puesto :

```
andreisudouser@andreidani-equipo:~/easy-rsa$ ./easyrsa build-ca nopass
```

Ahora tengo dos archivos que conforman los componentes públicos y privados de una entidad de certificación:

```
andreisudouser@andreidani-equipo:~/easy-rsa$ ls
easyrsa  openssl-easyrsa.cnf  pki  vars  vars.example  x509-types
andreisudouser@andreidani-equipo:~/easy-rsa$ ls pki/ca.crt
pki/ca.crt
andreisudouser@andreidani-equipo:~/easy-rsa$ ls pki/private/ca.key
pki/private/ca.key
```

2) Manda una peticion de firma de certificado CSR y firmala con la CA.

1) Instalo OpenSSL:

```
andreisudouser@andreidani-equipo:~$ sudo apt update
```

```
andreisudouser@andreidani-equipo:~$ sudo apt install openssl
```

2) Creo una CSR con openssl. Primero hay que generar una clave privada, y para ello necesito crear una nueva carpeta llamada practice-csr y generar la clave dentro de la misma. En vez de crear un certificado para identificar usuarios u otras CA, se crea esta solicitud para un server ficticio llamado sammy-server.

```
andreisudouser@andreidani-equipo:~$ mkdir ~/practice-csr
andreisudouser@andreidani-equipo:~$ cd ~/practice-csr/
andreisudouser@andreidani-equipo:~/practice-csr$ openssl genrsa -out sammy-server.key
```

```
andreisudouser@andreidani-equipo:~/practice-csr$ ls
sammy-server.key
```

Ahora ya tendría una clave privada y puedo crear la CSR y usar openssl. Se piden varios campos, se puede dejar en blanco poniendo un punto (.).

```
andreisudouser@andreidani-equipo:~/practice-csr$ openssl req -new -key sammy-server.key -out sammy-s
erver.req
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]:Madrid
Locality Name (eg, city) []:San Fernando de Henares
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Andreipeticion
Organizational Unit Name (eg, section) []:Andreipeticion2
Common Name (e.g. server FQDN or YOUR name) []:ANDREI DANIEL GRIGORE
Email Address []:aaaaaaaa@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:andreidani
An optional company name []:nombrecompania
andreisudouser@andreidani-equipo:~/practice-csr$ _
```

Se puede verificar lo que contiene el CSR con:

```
andreisudouser@andreidani-equipo:~/practice-csr$ openssl req -in sammy-server.req -noout -subject
subject=C = ES, ST = Madrid, L = San Fernando de Henares, O = Andreipeticion, OU = Andreipeticion2,
CN = ANDREI DANIEL GRIGORE, emailAddress = aaaaaaaa@gmail.com
```

3) Genero una solicitud de firma de certificado para un servidor ficticio Sammy-server copiando el archivo:

```
andreisudouser@andreidani-equipo:~/practice-csr$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:b5:f9:5b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 79293sec preferred_lft 79293sec
    inet6 fe80::a00:27ff:feb5:f95b/64 scope link
       valid_lft forever preferred_lft forever
andreisudouser@andreidani-equipo:~/practice-csr$ scp sammy-server.req andreisudouser@10.0.2.15:/tmp/
sammy-server.req
andreisudouser@10.0.2.15's password:
sammy-server.req                                      100% 1192     4.9MB/s   00:00
```

4) Para firmar la CSR ficticia importo la solicitud de certificado:

```
andreisudouser@andreidani-equipo:~$ ls
easy-rsa  practice-csr
andreisudouser@andreidani-equipo:~$ cd ~/easy-rsa
andreisudouser@andreidani-equipo:~/easy-rsa$ ./easyrsa import-eq /tmp/sammy-server.req sammy-server

Easy-RSA error:

Unknown command 'import-eq'. Run without commands for usage help.

andreisudouser@andreidani-equipo:~/easy-rsa$ ./easyrsa import-req /tmp/sammy-server.req sammy-server
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)

The request has been successfully imported with a short name of: sammy-server
You may now use this name to perform signing operations on this request.
```

5) Firmo la solicitud con el siguiente comando con la opción sign-req, seguida del tipo de solicitud y el nombre común incluido en la CSR. Como es una prueba practicando con un certificado para un servidor ficticio hay que usar el tipo de solicitud server:

```
andreisudouser@andreidani-equipo:~/easy-rsa$ ./easyrsa sign-req server sammy-server
Using SSL: openssl OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
    countryName               = ES
    stateOrProvinceName       = Madrid
    localityName              = San Fernando de Henares
    organizationName          = Andreipeticion
    organizationalUnitName    = Andreipeticion2
    commonName                = ANDREI DANIEL GRIGORE
    emailAddress              = aaaaaaaa@gmail.com


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
```

Ingreso la contraseña que introducí para la CA:

```
Using configuration from /home/andreisudouser/easy-rsa/pki/easy-rsa-4863.NzdLt7/tmp.tFm2tI
Enter pass phrase for /home/andreisudouser/easy-rsa/pki/private/ca.key:
40171373B97F0000:error:0700006C:configuration file routines:NCONF_get_string:no value:../crypto/con
/conf_lib.c:315:group=<NULL> name=unique_subject
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'ES'
stateOrProvinceName   :ASN.1 12:'Madrid'
localityName          :ASN.1 12:'San Fernando de Henares'
organizationName      :ASN.1 12:'Andreipeticion'
organizationalUnitName:ASN.1 12:'Andreipeticion2'
commonName            :ASN.1 12:'ANDREI DANIEL GRIGORE'
emailAddress          :IA5STRING:'aaaaaaaa@gmail.com'
Certificate is to be certified until Feb 19 21:41:50 2025 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /home/andreisudouser/easy-rsa/pki/issued/sammy-server.crt
```

Con el servidor de CA se firmó la CSR de sammy-server.req en el archivo ca.key. De ello sale el archivo Sammy-server.crt conteniendo la clave de cifrado pública del servidor que tengo de prueba incluída la firma del servidor de CA.
El objetivo es indicar que quienes confían en la CA también pueden hacerlo en el certificado de Sammy-server

Como es un servidor de prueba se ha usado otro procedimiento para poder probar esto, en un servidor real habría que distribuir los nuevos archivos Sammy-server.crt y ca.crt del servidor de CA al servidor remoto que realizó la solicitud de firma de certificado:

```
andreisudouser@andreidani-equipo:~/easy-rsa$ scp pki/issued/sammy-server.crt andreisudouser@10.0.2.1
5:/tmp
andreisudouser@10.0.2.15's password:
sammy-server.crt                                      100% 5001    19.0MB/s   00:00
andreisudouser@andreidani-equipo:~/easy-rsa$
andreisudouser@andreidani-equipo:~/easy-rsa$ scp pki/ca.crt andreisudouser@10.0.2.15:/tmp
andreisudouser@10.0.2.15's password:
ca.crt                                                100% 1204     5.1MB/s   00:00
andreisudouser@andreidani-equipo:~/easy-rsa$
```