Module-5: Networking and Security

Demo Document - 6

DEMO-6: Securing the Daemon

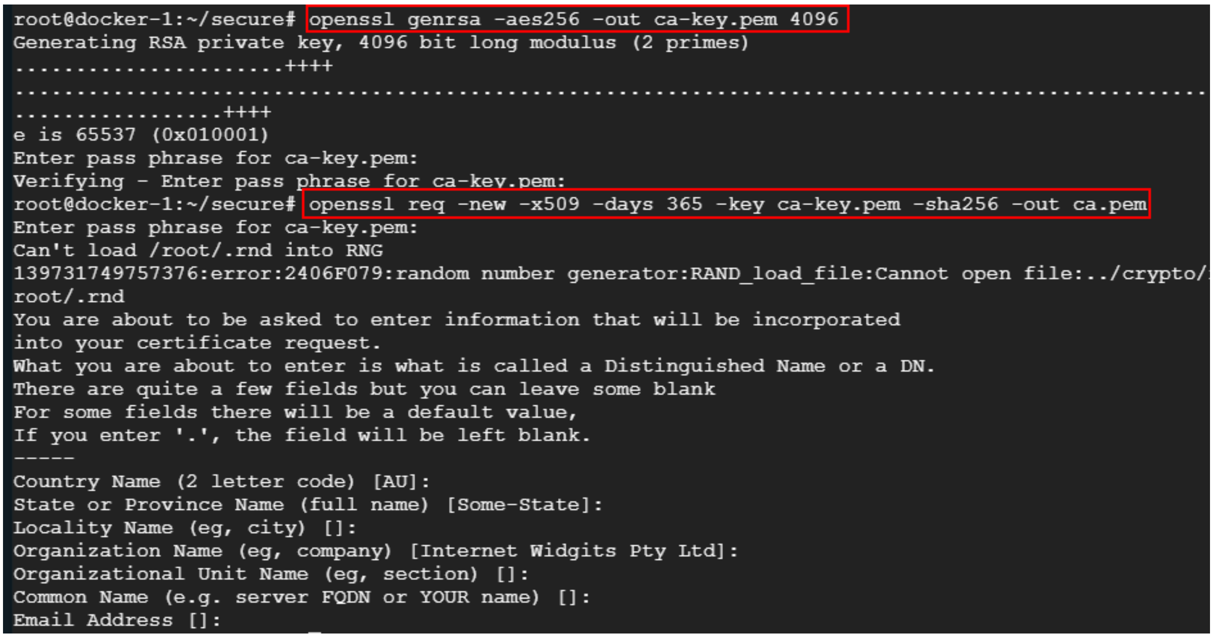
Note: All commands are executed as root.

Host Machine

1. Generate public and private CA keys using openssl

```
$ openssl genrsa -aes256 -out ca-key.pem 4096
```

```
$ openssl genrsa -aes256 -out ca-key.pem 4096
```



```
root@docker-1:~/secure# openssl genrsa -aes256 -out ca-key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
......................++++
.................................................................................................
.................++++
e is 65537 (0x010001)
Enter pass phrase for ca-key.pem:
Verifying - Enter pass phrase for ca-key.pem:
root@docker-1:~/secure# openssl req -new -x509 -days 365 -key ca-key.pem -sha256 -out ca.pem
Enter pass phrase for ca-key.pem:
Can't load /root/.rnd into RNG
139731749757376:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/
root/.rnd
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]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

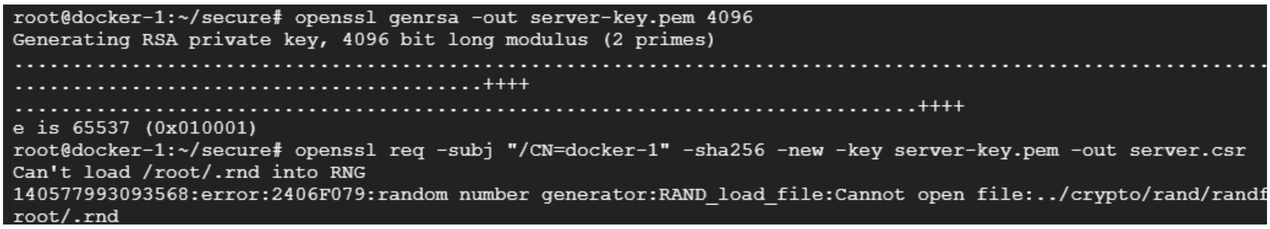
2. Create a server key and a Certificate Signing Request (CSR)

```
$ openssl genrsa -out server-key.pem 4096
```

Replace the $HOST with your hostname

```
$ openssl req -subj "/CN=$HOST" -sha256 -new -key server-key.pem -out server.csr
```

This command may give you a .rnd error. Ignore it



```
root@docker-1:~/secure# openssl genrsa -out server-key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.........................................................................................................
......................................++++
..........................................................................++++
e is 65537 (0x010001)
root@docker-1:~/secure# openssl req -subj "/CN=docker-1" -sha256 -new -key server-key.pem -out server.csr
Can't load /root/.rnd into RNG
140577993093568:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/rand/randf
root/.rnd
```

3. Sign the public key with the CA and set key extended usage to serverAuth

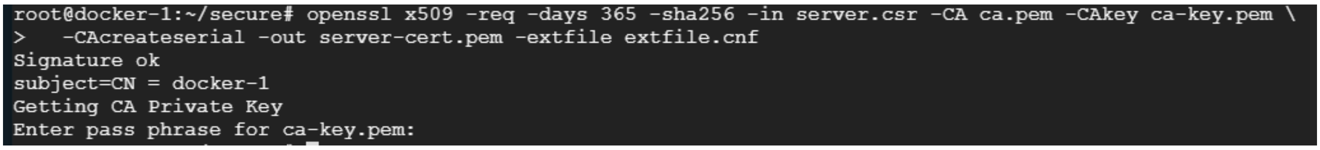
```
$ echo subjectAltName = DNS:$HOST,IP:<hostIP>,IP:127.0.0.1 >> extfile.cnf

$ echo extendedKeyUsage = serverAuth >> extfile.cnf
```

4. Now, generate the signed certificate

```
$ openssl x509 -req -days 365 -sha256 -in server.csr -CA ca.pem -CAkey ca-
key.pem \

  -CAcreateserial -out server-cert.pem -extfile extfile.cnf
```

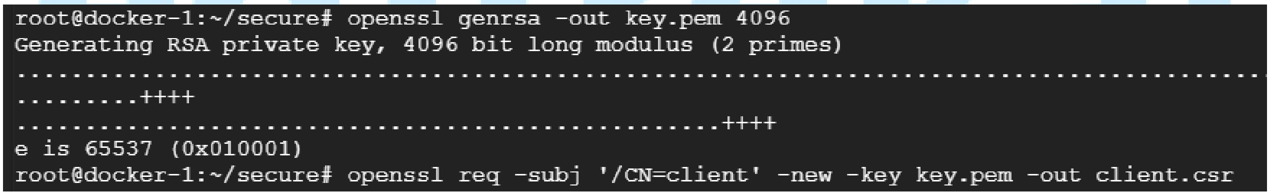


```
root@docker-1:~/secure# openssl x509 -req -days 365 -sha256 -in server.csr -CA ca.pem -CAkey ca-key.pem \
>   -CAcreateserial -out server-cert.pem -extfile extfile.cnf
Signature ok
subject=CN = docker-1
Getting CA Private Key
Enter pass phrase for ca-key.pem:
```

5. Create client key and CSR

```
$ openssl genrsa -out key.pem 4096

$ openssl req -subj '/CN=client' -new -key key.pem -out client.csr
```



```
root@docker-1:~/secure# openssl genrsa -out key.pem 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....................................................................................................
.........++++
.......................................................++++
e is 65537 (0x010001)
root@docker-1:~/secure# openssl req -subj '/CN=client' -new -key key.pem -out client.csr
```

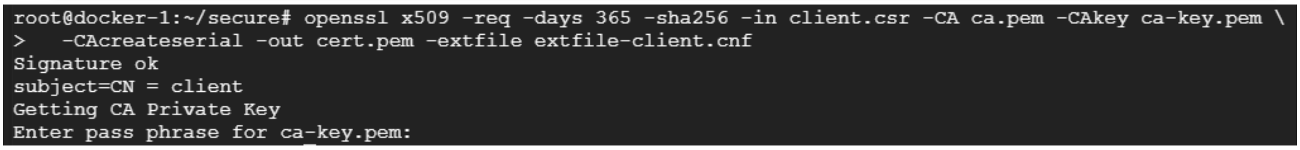
6. Create a new extensions config file to make the key suitable for client authentication

```
$ echo extendedKeyUsage = clientAuth > extfile-client.cnf
```

7. Generate the signed certificate

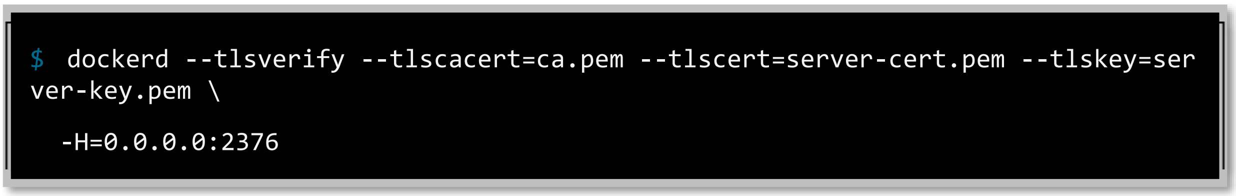
```
$ openssl x509 -req -days 365 -sha256 -in client.csr -CA ca.pem -CAkey ca-key.p
em \

  -CAcreateserial -out cert.pem -extfile extfile-client.cnf
```



```
root@docker-1:~/secure# openssl x509 -req -days 365 -sha256 -in client.csr -CA ca.pem -CAkey ca-key.pem \
>   -CAcreateserial -out cert.pem -extfile extfile-client.cnf
Signature ok
subject=CN = client
Getting CA Private Key
Enter pass phrase for ca-key.pem:
```

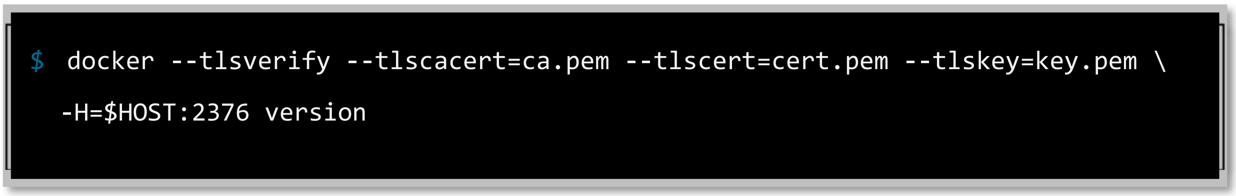
8. Now you have the certificates which allow your Docker daemon to accept connections only from clients trusted by your CA



```
$ dockerd --tlsverify --tlscacert=ca.pem --tlscert=server-cert.pem --tlskey=ser
ver-key.pem \

  -H=0.0.0.0:2376
```

9. Copy your CA certificate, server certificate and your client certificate to the client machine and run the following command



```
$ docker --tlsverify --tlscacert=ca.pem --tlscert=cert.pem --tlskey=key.pem \

  -H=$HOST:2376 version
```

Replace $HOST with DNS of your docker host