Managing Certificates with OpenSSL in Linux

Scenario

In this activity, you will work with OpenSSL to manage certificates. You will generate certificates, generate a certificate signing request, and convert certificate formats. These are all tasks administrators are responsible for when managing web servers, email servers, and other devices.

Objectives

This activity is designed to test your understanding of and ability to apply content examples in the following CompTIA Security+ objectives:

- 3.9 Given a scenario, implement public key infrastructure.

Lab

- Kali VM
- pfSense VM

Task 1

Use basic OpenSSL commands

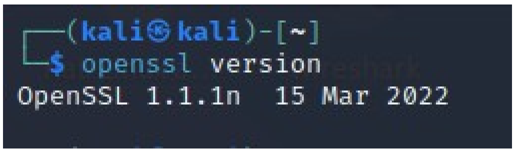
In the first part of this activity, you will use basic OpenSSL commands to confirm the program version. You will also create a storage directory for the keys that you will generate.

1. Sign in to the **Kali** VM as **kali** using **Pa$$w0rd** as the password.

2. Open the **Terminal** from the menu at the top of the Desktop.

3. To check OpenSSL version, type the following command, and then press **ENTER**:

```
openssl version
```



```
┌──(kali㉿kali)-[~]
└─$ openssl version
OpenSSL 1.1.1n  15 Mar 2022
```

Figure 1.1 – OpenSSL version.

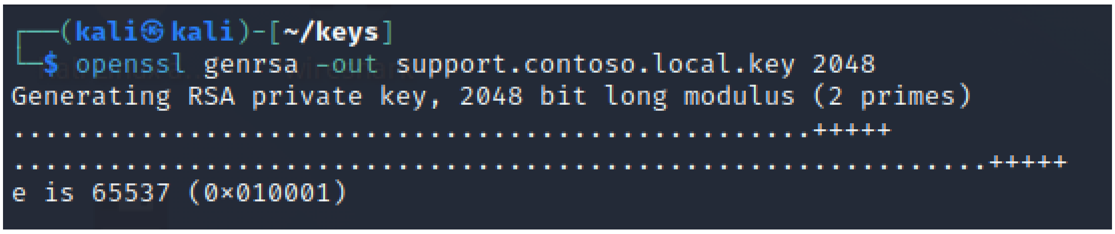
4. Run the following command to create a directory named **keys** in the kali user's home directory.

```
mkdir keys
```

5. Use the **cd** command to change to the **keys** directory.

6. Generate an asymmetric encryption RSA key pair and extract the public portion to prepare to create a certificate signing request (which occurs below). Type the following command, then press **ENTER**:

```
openssl genrsa -out support.contoso.local.key 2048
```

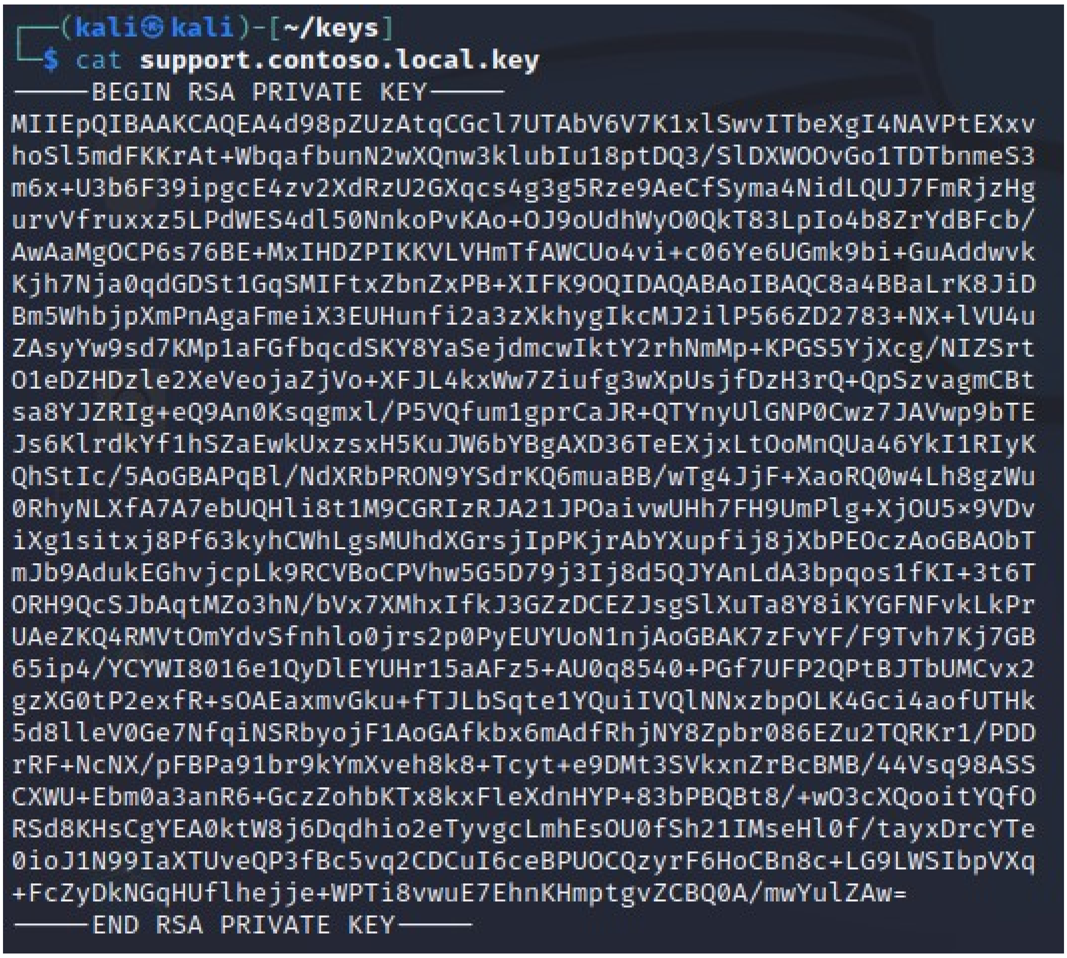


```
┌──(kali㉿kali)-[~/keys]
└─$ openssl genrsa -out support.contoso.local.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
..............................................+++++
..............................................................+++++
e is 65537 (0×010001)
```

Figure 1.2 – Generating an RSA private key with OpenSSL.

7. Run the following command to display the private key:

```
cat support.contoso.local.key
```



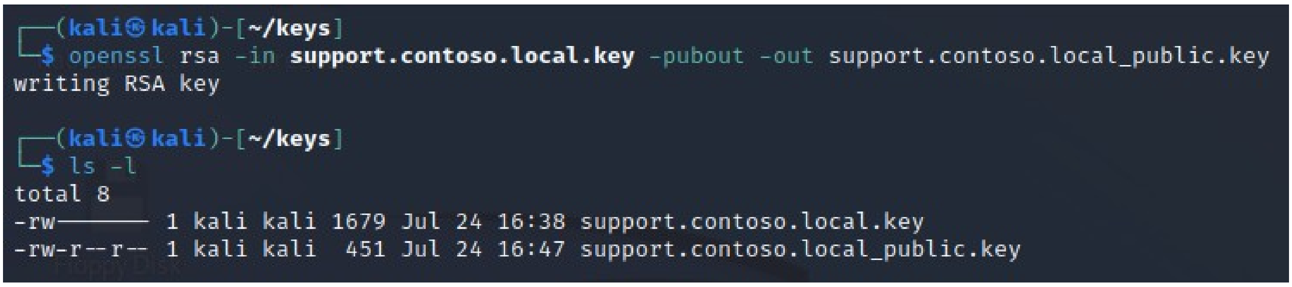
```
┌──(kali㉿kali)-[~/keys]
└─$ cat support.contoso.local.key
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA4d98pZUzAtqCGcl7UTAbV6V7K1xlSwvITbeXgI4NAVPtEXxv
hoSl5mdFKKrAt+WbqafbunN2wXQnw3klubIu18ptDQ3/SlDXWOOvGo1TDTbnmeS3
m6x+U3b6F39ipgcE4zv2XdRzU2GXqcs4g3g5Rze9AeCfSyma4NidLQUJ7FmRjzHg
urvVfruxxz5LPdWES4dl50NnkoPvKAo+OJ9oUdhWyO0QkT83LpIo4b8ZrYdBFcb/
AwAaMgOCP6s76BE+MxIHDZPIKKVLVHmTfAWCUo4vi+c06Ye6UGmk9bi+GuAddwvk
Kjh7Nja0qdGDSt1GqSMIFtxZbnZxPB+XIFK9OQIDAQABAoIBAQC8a4BBaLrK8JiD
Bm5WhbjpXmPnAgaFmeiX3EUHunfi2a3zXkhygIkcMJ2ilP566ZD2783+NX+lVU4u
ZAsyYw9sd7KMp1aFGfbqcdSKY8YaSejdmcwIktY2rhNmMp+KPGS5YjXcg/NIZSrt
O1eDZHDzle2XeVeojaZjVo+XFJL4kxWw7Ziufg3wXpUsjfDzH3rQ+QpSzvagmCBt
sa8YJZRIg+eQ9An0Ksqgmxl/P5VQfum1gprCaJR+QTYnyUlGNP0Cwz7JAVwp9bTE
Js6KlrdkYf1hSZaEwkUxzsxH5KuJW6bYBgAXD36TeEXjxLtOoMnQUa46YkI1RIyK
QhStIc/5AoGBAPqBl/NdXRbPRON9YSdrKQ6muaBB/wTg4JjF+XaoRQ0w4Lh8gzWu
0RhyNLXfA7A7ebUQHli8t1M9CGRIzRJA21JPOaivwUHh7FH9UmPlg+XjOU5×9VDv
iXg1sitxj8Pf63kyhCWhLgsMUhdXGrsjIpPKjrAbYXupfij8jXbPEOczAoGBAObT
mJb9AdukEGhvjcpLk9RCVBoCPVhw5G5D79j3Ij8d5QJYAnLdA3bpqos1fKI+3t6T
ORH9QcSJbAqtMZo3hN/bVx7XMhxIfkJ3GZzDCEZJsgSlXuTa8Y8iKYGFNFvkLkPr
UAeZKQ4RMVtOmYdvSfnhlo0jrs2p0PyEUYUoN1njAoGBAK7zFvYF/F9Tvh7Kj7GB
65ip4/YCYWI8016e1QyDlEYUHr15aAFz5+AU0q8540+PGf7UFP2QPtBJTbUMCvx2
gzXG0tP2exfR+sOAEaxmvGku+fTJLbSqte1YQuiIVQlNNxzbpOLK4Gci4aofUTHk
5d8lleV0Ge7NfqiNSRbyojF1AoGAfkbx6mAdfRhjNY8Zpbr086EZu2TQRKr1/PDD
rRF+NcNX/pFBPa91br9kYmXveh8k8+Tcyt+e9DMt3SVkxnZrBcBMB/44Vsq98ASS
CXWU+Ebm0a3anR6+GczZohbKTx8kxFleXdnHYP+83bPBQBt8/+wO3cXQooitYQfO
RSd8KHsCgYEA0ktW8j6Dqdhio2eTyvgcLmhEsOU0fSh21IMseHl0f/tayxDrcYTe
0ioJ1N99IaXTUveQP3fBc5vq2CDCuI6ceBPUOCQzyrF6HoCBn8c+LG9LWSIbpVXq
+FcZyDkNGqHUflhejje+WPTi8vwuE7EhnKHmptgvZCBQ0A/mwYulZAw=
-----END RSA PRIVATE KEY-----
```

Figure 1.3 – The Private key file.

8. Extract the public key file to a file for export with a CSR.

```
openssl rsa -in support.contoso.local.key -pubout -out
support.contoso.local_public.key
```

9. Use the **ls -l** command to display the two key files that you have created so far.



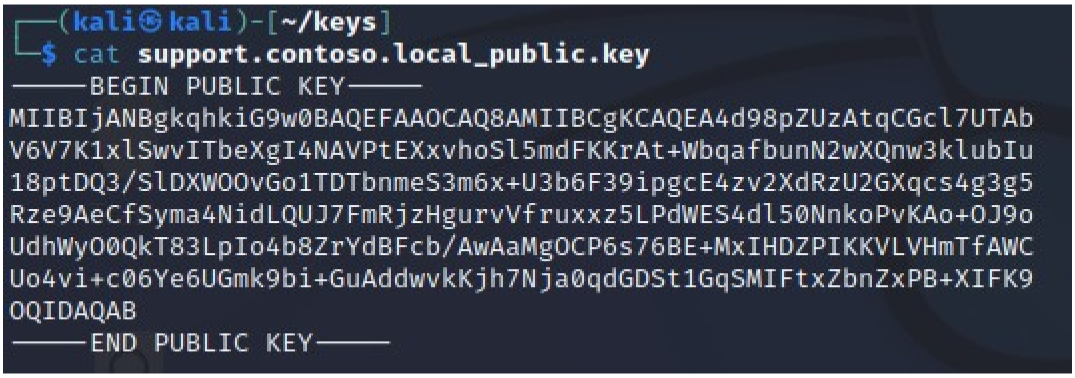
```
┌──(kali㉿kali)-[~/keys]
└─$ openssl rsa -in support.contoso.local.key -pubout -out support.contoso.local_public.key
writing RSA key

┌──(kali㉿kali)-[~/keys]
└─$ ls -l
total 8
-rw------- 1 kali kali 1679 Jul 24 16:38 support.contoso.local.key
-rw-r--r-- 1 kali kali  451 Jul 24 16:47 support.contoso.local_public.key
```

Figure 1.4 – The two key files.

10. Display the public key file:

```
cat support.contoso.local_public.key
```



```
┌──(kali㉿kali)-[~/keys]
└─$ cat support.contoso.local_public.key
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4d98pZUzAtqCGcl7UTAb
V6V7K1xlSwvITbeXgI4NAVPtEXxvhoSl5mdFKKrAt+WbqafbunN2wXQnw3klubIu
18ptDQ3/SlDXWOOvGo1TDTbnmeS3m6x+U3b6F39ipgcE4zv2XdRzU2GXqcs4g3g5
Rze9AeCfSyma4NidLQUJ7FmRjzHgurvVfruxxz5LPdWES4dl50NnkoPvKAo+OJ9o
UdhWyO0QkT83LpIo4b8ZrYdBFcb/AwAaMgOCP6s76BE+MxIHDZPIKKVLVHmTfAWC
Uo4vi+c06Ye6UGmk9bi+GuAddwvkKjh7Nja0qdGDSt1GqSMIFtxZbnZxPB+XIFK9
OQIDAQAB
-----END PUBLIC KEY-----
```

Figure 1.5 – The public key file.

Task 2

Generate a certificate signing request

Generate a web site certificate signing request that could be sent to a certificate authority (CA).

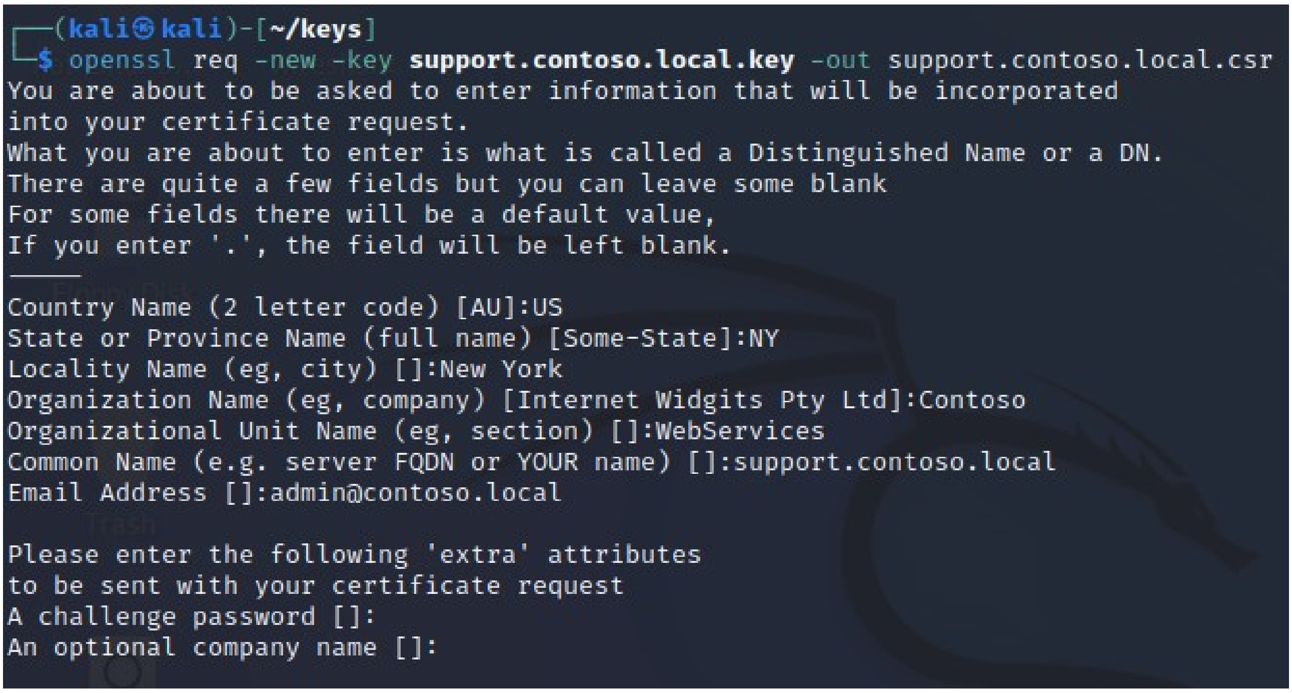
1. Generate a certificate signing request. Type the following command, then press **ENTER**:

```
openssl req -new -key support.contoso.local.key -out
support.contoso.local.csr
```

2. Provide the following answers to the prompts:

- Country Code: *your country*
- State or Province Name: *your state or province*
- Locality Name: *your city*
- Organization Name: Contoso
- Organizational Unit Name: WebServices
- Common Name: support.contoso.local
- Email Address: admin@contoso.local

3. When prompted to enter a challenge password and an optional company name, press **ENTER**.



```
┌──(kali㉿kali)-[~/keys]
└─$ openssl req -new -key support.contoso.local.key -out support.contoso.local.csr
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]:US
State or Province Name (full name) [Some-State]:NY
Locality Name (eg, city) []:New York
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Contoso
Organizational Unit Name (eg, section) []:WebServices
Common Name (e.g. server FQDN or YOUR name) []:support.contoso.local
Email Address []:admin@contoso.local

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

Figure 2.1 – Generating a Certificate Signing Request.

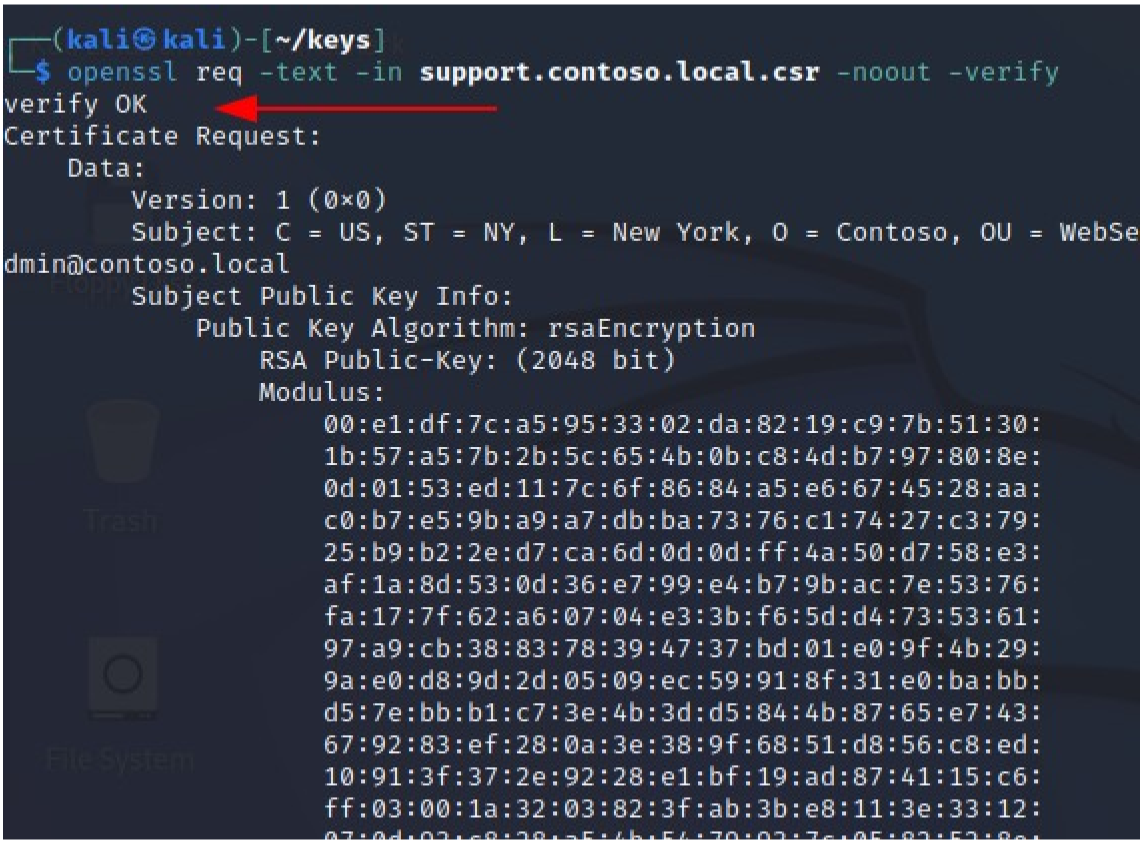
This OpenSSL command generates a certificate signing request on behalf of the Apache web server service for a web site.

4. Run the **ls -l** command to display the .csr file.

5. Verify the certificate request. Type the following command, then press **ENTER**:

```
openssl req -text -in support.contoso.local.csr -noout -verify
```

This OpenSSL command verifies the certificate request.

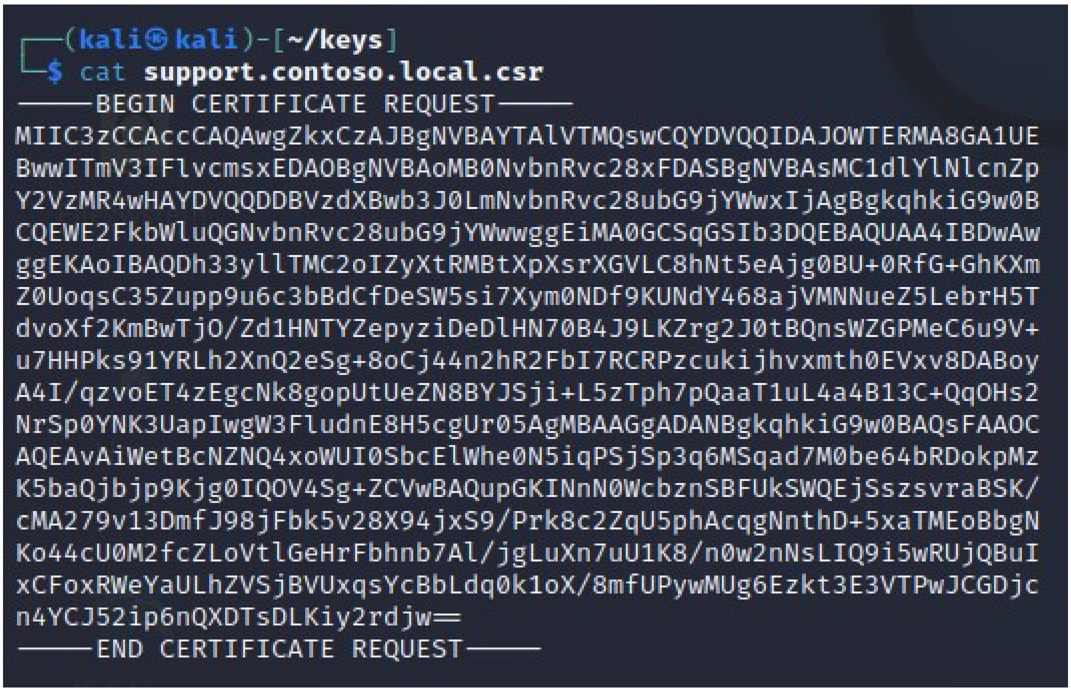


```
┌──(kali㉿kali)-[~/keys]
└─$ openssl req -text -in support.contoso.local.csr -noout -verify
verify OK
Certificate Request:
    Data:
        Version: 1 (0×0)
        Subject: C = US, ST = NY, L = New York, O = Contoso, OU = WebSe
dmin@contoso.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:e1:df:7c:a5:95:33:02:da:82:19:c9:7b:51:30:
                    1b:57:a5:7b:2b:5c:65:4b:0b:c8:4d:b7:97:80:8e:
                    0d:01:53:ed:11:7c:6f:86:84:a5:e6:67:45:28:aa:
                    c0:b7:e5:9b:a9:a7:db:ba:73:76:c1:74:27:c3:79:
                    25:b9:b2:2e:d7:ca:6d:0d:0d:ff:4a:50:d7:58:e3:
                    af:1a:8d:53:0d:36:e7:99:e4:b7:9b:ac:7e:53:76:
                    fa:17:7f:62:a6:07:04:e3:3b:f6:5d:d4:73:53:61:
                    97:a9:cb:38:83:78:39:47:37:bd:01:e0:9f:4b:29:
                    9a:e0:d8:9d:2d:05:09:ec:59:91:8f:31:e0:ba:bb:
                    d5:7e:bb:b1:c7:3e:4b:3d:d5:84:4b:87:65:e7:43:
                    67:92:83:ef:28:0a:3e:38:9f:68:51:d8:56:c8:ed:
                    10:91:3f:37:2e:92:28:e1:bf:19:ad:87:41:15:c6:
                    ff:03:00:1a:32:03:82:3f:ab:3b:e8:11:3e:33:12:
```

Figure 2.2 – Verifying the Certificate request.

6. The certificate signing request must be sent to the certificate authority using PEM format. Run the following command to display the CSR file in this format:

```
cat support.contoso.local.csr
```



```
┌──(kali㉿kali)-[~/keys]
└─$ cat support.contoso.local.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIC3zCCAccCAQAwgZkxCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJOWTERMA8GA1UE
BwwITmV3IFlvcmsxEDAOBgNVBAoMB0NvbnRvc28xFDASBgNVBAsMC1dlYlNlcnZp
Y2VzMR4wHAYDVQQDDBVzdXBwb3J0LmNvbnRvc28ubG9jYWwxIjAgBgkqhkiG9w0B
CQEWE2FkbWluQGNvbnRvc28ubG9jYWwwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQDh33yllTMC2oIZyXtRMBtXpXsrXGVLC8hNt5eAjg0BU+0RfG+GhKXm
Z0UoqsC35Zupp9u6c3bBdCfDeSW5si7Xym0NDf9KUNdY468ajVMNNueZ5LebrH5T
dvoXf2KmBwTjO/Zd1HNTYZepyziDeDlHN70B4J9LKZrg2J0tBQnsWZGPMeC6u9V+
u7HHPks91YRLh2XnQ2eSg+8oCj44n2hR2FbI7RCRPzcukijhvxmth0EVxv8DABoy
A4I/qzvoET4zEgcNk8gopUtUeZN8BYJSji+L5zTph7pQaaT1uL4a4B13C+QqOHs2
NrSp0YNK3UapIwgW3FludnE8H5cgUr05AgMBAAGgADANBgkqhkiG9w0BAQsFAAOC
AQEAvAiWetBcNZNQ4xoWUI0SbcElWhe0N5iqPSjSp3q6MSqad7M0be64bRDokpMz
K5baQjbjp9Kjg0IQOV4Sg+ZCVwBAQupGKINnN0WcbznSBFUkSWQEjSszsvraBSK/
cMA279v13DmfJ98jFbk5v28X94jxS9/Prk8c2ZqU5phAcqgNnthD+5xaTMEoBbgN
Ko44cU0M2fcZLoVtlGeHrFbhnb7Al/jgLuXn7uU1K8/n0w2nNsLIQ9i5wRUjQBuI
xCFoxRWeYaULhZVSjBVUxqsYcBbLdq0k1oX/8mfUPywMUg6Ezkt3E3VTPwJCGDjc
n4YCJ52ip6nQXDTsDLKiy2rdjw==
-----END CERTIFICATE REQUEST-----
```

Figure 2.3 – Output from the Certificate Signing Request.

NOTE: The .local extension is an example extension for a non-routable domain. In real production environments you would use one of the Top Level Domain extensions like .com, .net, etc.

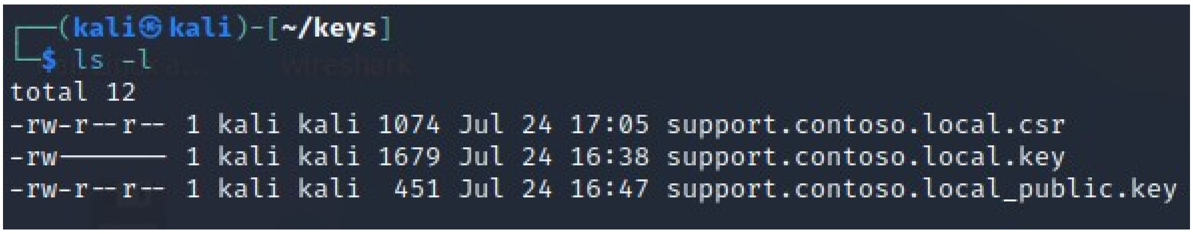
NOTE: The entire contents of the output must be copied into the certificate request interface (this interface varies by CA vendor). The header -----BEGIN CERTIFICATE REQUEST----- and trailer -----END CERTIFICATE REQUEST----- must be included.

Task 3

Convert certificate format

As we don’t have a CA available to sign the request, for the next part of the exercise we’ll generate a self-signed certificate with a new key, overwriting the old one.

1. Run the **ls -l** command and observe that there are three files in the directory.



```
┌──(kali㉿kali)-[~/keys]
└─$ ls -l
total 12
-rw-r--r-- 1 kali kali 1074 Jul 24 17:05 support.contoso.local.csr
-rw------- 1 kali kali 1679 Jul 24 16:38 support.contoso.local.key
-rw-r--r-- 1 kali kali  451 Jul 24 16:47 support.contoso.local_public.key
```

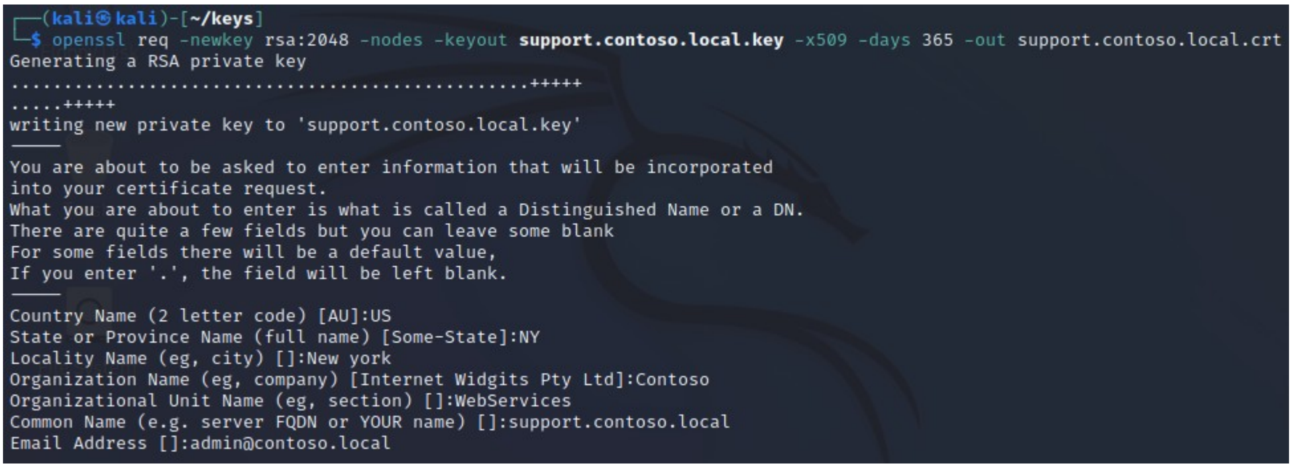
Figure 3.1 – Listing the keys directory contents.

2. Generate a self-signed certificate:

```
openssl req -newkey rsa:2048 -nodes -keyout support.consoto.local.key -
x509 -days 365 -out support.contoso.local.crt
```

3. Provide the following answers to the prompts:

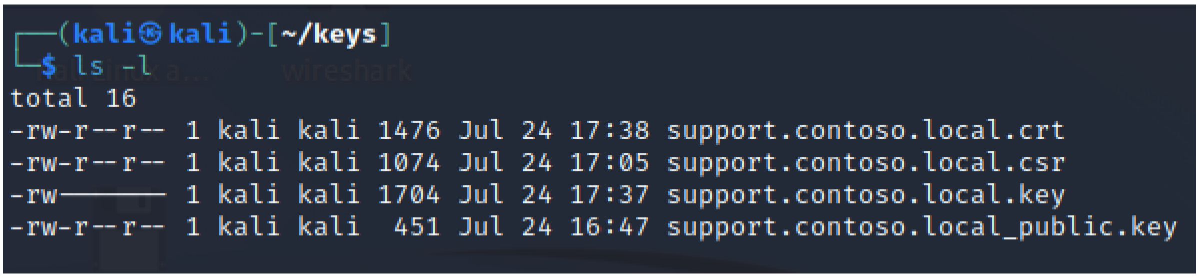
- Country Code: *your country*
- State or Province Name: *your state or province*
- Locality Name: *your city*
- Organization Name: Contoso
- Organizational Unit Name: WebServices
- Common Name: support.contoso.local
- Email Address: admin@contoso.local



```
┌──(kali㉿kali)-[~/keys]
└─$ openssl req -newkey rsa:2048 -nodes -keyout support.contoso.local.key -x509 -days 365 -out support.contoso.local.crt
Generating a RSA private key
.................................................+++++
.....+++++
writing new private key to 'support.contoso.local.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]:US
State or Province Name (full name) [Some-State]:NY
Locality Name (eg, city) []:New york
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Contoso
Organizational Unit Name (eg, section) []:WebServices
Common Name (e.g. server FQDN or YOUR name) []:support.contoso.local
Email Address []:admin@contoso.local
```

Figure 3.2 – Generating a Self-Signed certificate.

4. Run the **ls -l** command again and observe that there are now four files in the directory. A new .crt file has been created.



```
┌──(kali㉿kali)-[~/keys]
└─$ ls -l
total 16
-rw-r--r-- 1 kali kali 1476 Jul 24 17:38 support.contoso.local.crt
-rw-r--r-- 1 kali kali 1074 Jul 24 17:05 support.contoso.local.csr
-rw─────── 1 kali kali 1704 Jul 24 17:37 support.contoso.local.key
-rw-r--r-- 1 kali kali  451 Jul 24 16:47 support.contoso.local_public.key
```

Figure 3.3 – Listing the keys directory contents.

Task 4

Merge the .key and .crt files (the non-Windows PEM format) into a .pfx file (the Windows PKCS#12 format)

Convert the certificate you generated to a format accepted by the Windows operating system.

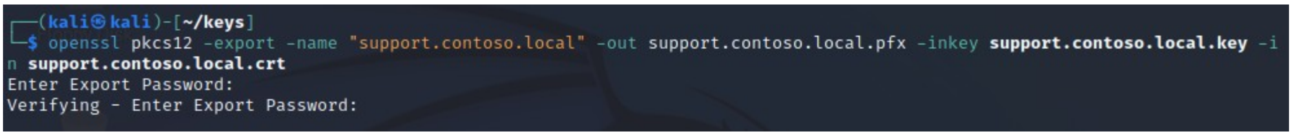
Certificates are often issued by CAs in the PEM format. Windows servers often use the PKCS#12 format, where keys are merged into a single file. The PKCS#12 format is an archival file that stores both the certificate and the private key. This format is useful for migrating certificates and keys from one system to another as it contains all the necessary files. PKCS#12 files use either the .pfx or .p12 file extension.

Use the following command to convert your PEM key and certificate into the PKCS#12 format (i.e., a single .pfx file):

1. Type the following command to convert the files, then press **ENTER**:

```
openssl pkcs12 -export -name “support.contoso.local” -out
support.contoso.local.pfx -inkey support.contoso.local.key -in
support.contoso.local.crt
```

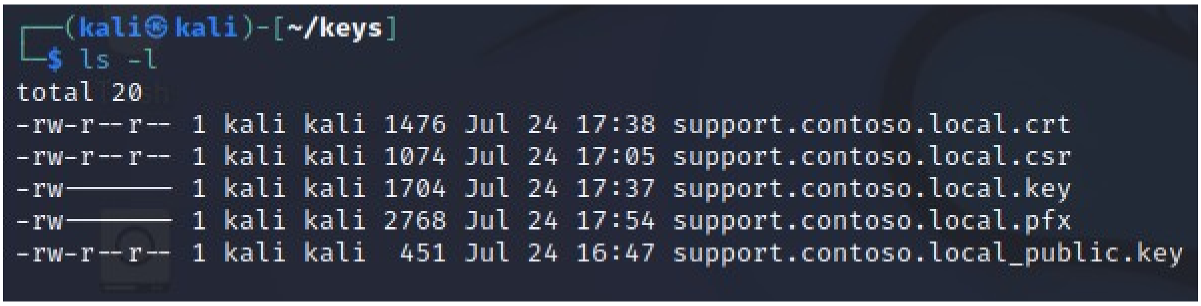
2. When prompted select **ENTER** to skip defining an **Export Password**.



```
┌──(kali㉿kali)-[~/keys]
└─$ openssl pkcs12 -export -name "support.contoso.local" -out support.contoso.local.pfx -inkey support.contoso.local.key -i
n support.contoso.local.crt
Enter Export Password:
Verifying - Enter Export Password:
```

Figure 4.1 – OpenSSL converting to PKCS#12 format.

3. Run the **ls -l** command and observe that there are now five files in the directory. A new .pfx file has been created.



```
┌──(kali㉿kali)-[~/keys]
└─$ ls -l
total 20
-rw-r--r-- 1 kali kali 1476 Jul 24 17:38 support.contoso.local.crt
-rw-r--r-- 1 kali kali 1074 Jul 24 17:05 support.contoso.local.csr
-rw------- 1 kali kali 1704 Jul 24 17:37 support.contoso.local.key
-rw------- 1 kali kali 2768 Jul 24 17:54 support.contoso.local.pfx
-rw-r--r-- 1 kali kali  451 Jul 24 16:47 support.contoso.local_public.key
```

Figure 4.2 – Using ls -l to list the contents of the keys directory.