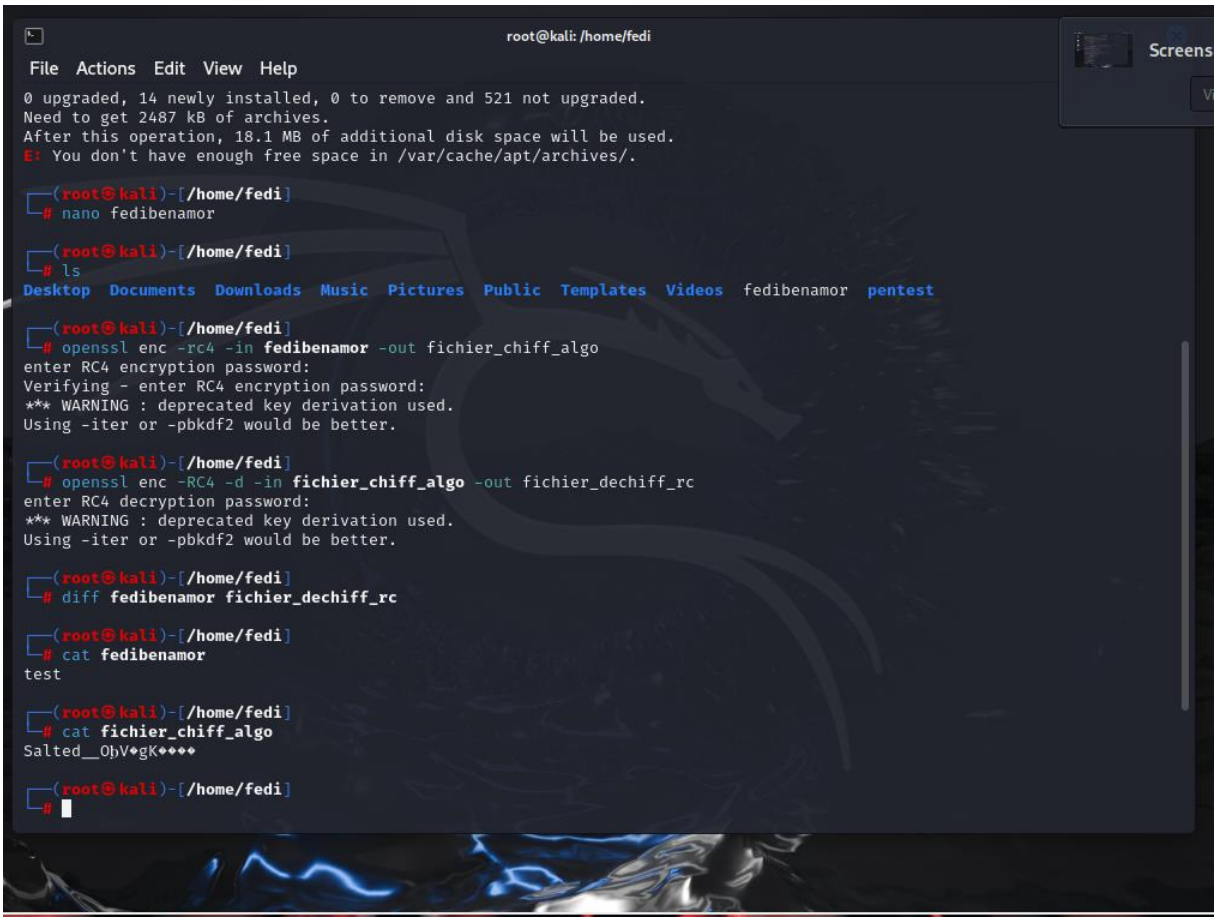
Rendu

1. CHIFFREMENT SYMETRIQUE

L'algorithme RC4



L'algorithme DES

```
(root㉿kali)-[/home/fedi]
# openssl enc -des -pbkdf2 -in fedibenamor -out fichier_chiff_des -k 0123456789ABCDEF
```

```
(root㉿kali)-[/home/fedi]
# cat fichier_chiff_des
Salted__◆◆◆-◆◆~/.◆*/◆
```

- Quelle est la commande qui permet de déchiffrer fichier_chiff_des ?

```
(root㉿kali)-[/home/fedi]
# openssl enc -des -d -pbkdf2 -in fichier_chiff_des -out fichier_dechiff_des -k 0123456789ABCDEF
```

- Vérifier que le fichier déchiffré est identique au fichier initial.

```
(root㉿kali)-[/home/fedi]
# diff fedibenamor fichier_dechiff_des

(root㉿kali)-[/home/fedi]
# cat fichier_dechiff_des
test
```

```
┌──(root㉿kali)-[/home/fedi]
└─# diff fedibenamor fichier_chiff_des
1c1
< test
---
> Salted__◆◆◆-◆◆~/.◆*/◆
\ No newline at end of file
```

2. CHIFFREMENT ASYMETRIQUE

- Générer une paire de clés de taille 1024 bits et stocker-la dans le fichier rsakey.pem

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl genrsa -out rsakey.pem 1024
```

- Afficher le fichier en utilisant la commande cat. Qu'est ce que vous remarquez ?

```
┌──(root㉿kali)-[/home/fedi]
└─# cat rsakey.pem
-----BEGIN PRIVATE KEY-----
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAMPlNAjHW66xfFnz
I1bSrbo1Mz7lub0xmBUK9BCzq2xPPIpoa0Wq4QdAnDX6jleszaVG1rMAD6gQQBo7
adfIbA6QTPBP0ayjewo0tWdHFoULvaL7VIRn9DUynRDXnTqwTIIlQPAN7CJ/vnGz
4/RlRjSUXtATXgN0CAMiIRn+y3K7AgMBAAECgYEAvK3/71CtQtaakjy4VgD36bki
EbvMqaih9eql54xWWSz8f8dRQznyIj0d+HPLNDExpbZURvaRHewBAgMlz/uv7qmz
vzOftsvfTdJHH1IoHYugYTQGdyNgYMlJ7kQIdQhSAz3siDWrTO3KfVupGPoe9tta
4HySUBB6H52RKrjNAfkCQQD2lPw4rtOGCXA+OVodbuYVTF9OEaG1Vr/MbVFVmnli
TgFo84Puv0XmBLDCC+tuPHslBPvjKUY74ikFbRbJRta9AkEAy2Cd+QpA3tVZeQii
ra2bseZEs4MjKFvEB60zcdehxPjrjTCTEyLf3WdAI5mRiBDH+RAjxN4×1Ys+umWd
+2gi1wJAZ4k7NpnB22APLzFpsz4jggHlaqRgoAHi2vVz11vbZ/mVAfpuBUtvEBwc
E/5pRopCstKa6VElWnOmAc9CaBVEiQJAIyaExylFgv3+49NhQoFR+pJg52HP7sbF
f8oorRFmzUN0esedc95AMOUKidyd8ZVs6pgmhn0cwdq6Mb8kafJaqwJAFInO42ku
Uzm4YybN50Wd9XvNMl40P5jXsr6exfYyNFSTPovQ7atRjwlTUXzUFqqOcqjt6lLX
Jwy8PiiR4keuVw==
-----END PRIVATE KEY-----
```

- Une façon de visualiser les clés en format complet est d'utiliser la commande rsa. Afficher alors les clés en format hexadécimal, en supprimant la sortie normalement produite par l'instruction rsa.

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl rsa -in rsakey.pem -text -noout
Private-Key: (1024 bit, 2 primes)
modulus:
    00:c3:e5:34:08:c7:5b:ae:b1:7c:59:f3:23:56:d2:
    ad:ba:35:33:3e:e5:b9:bd:31:98:15:0a:f4:10:b3:
    ab:6c:4f:3c:8a:68:6b:45:aa:e1:07:40:9c:35:fa:
    8e:57:ac:cd:a5:46:d6:b3:00:0f:a8:10:40:1a:3b:
    69:d7:c8:6c:0e:90:4c:f0:4f:d1:ac:a3:7b:0a:34:
    b5:67:47:16:85:0b:bd:a2:fb:54:84:67:f4:35:32:
    9d:10:d7:9d:3a:b0:4c:82:25:40:f0:0d:ec:22:7f:
    be:71:b3:e3:f4:65:46:34:94:5e:d0:13:5e:03:74:
    08:03:22:21:19:fe:cb:72:bb
publicExponent: 65537 (0x10001)
privateExponent:
    00:bc:ad:ff:ef:50:ad:42:d6:9a:92:3c:b8:56:00:
    f7:e9:b9:22:11:bb:cc:a9:a8:a1:f5:ea:a5:e7:8c:
    56:59:2c:fc:7f:c7:51:43:39:f2:22:3d:1d:f8:73:
    cb:34:31:31:a5:b6:54:46:f6:91:1d:ec:01:02:03:
    25:cf:fb:af:ee:a9:b3:bf:33:9f:b6:cb:df:4d:d2:
    47:1f:52:28:1d:8b:a0:61:34:06:77:23:60:60:c9:
    49:ee:44:08:75:08:52:03:3d:ec:88:35:ab:4c:ed:
    ca:7d:5b:a9:18:fa:1e:f6:db:5a:e0:7c:92:50:10:
    7a:1f:9d:91:2a:b8:cd:01:f9
prime1:
    00:f6:94:fc:38:ae:d3:86:09:70:3e:39:5a:1d:6e:
    e6:15:4c:5f:4e:11:a1:b5:56:bf:cc:6d:51:55:9a:
    79:62:4e:01:68:f3:83:ee:bf:45:e6:04:b0:c2:0b:
    eb:6e:3c:7b:25:04:fb:e3:29:46:3b:e2:29:05:6d:
    16:c9:46:d6:bd
prime2:
    00:cb:60:9d:f9:0a:40:de:d5:59:79:08:a2:ad:ad:
    9b:b1:e6:44:b3:83:23:28:5b:c4:07:ad:33:71:d7:
    a1:c4:f8:eb:8d:30:93:13:22:df:dd:67:40:23:99:
    91:88:10:c7:f9:10:23:c4:de:31:d5:8b:3e:ba:65:
    9d:fb:68:22:d7
exponent1:
    67:89:3b:36:99:c1:db:60:0f:2f:31:69:b3:3e:23:
    82:01:e5:6a:a4:60:a0:01:e2:da:f5:73:d7:5b:db:
    67:f9:95:01:fa:6e:05:4b:6f:10:1c:1c:13:fe:69:
    46:8a:42:b2:d2:9a:e9:51:25:5a:73:a6:01:cf:42:
    68:15:44:89
exponent2:
    23:26:84:c7:29:45:82:fd:fe:e3:d3:61:42:81:51:
    fa:92:60:e7:61:cf:ee:c6:c5:7f:ca:28:ad:11:66:
    cd:43:74:7a:c7:9d:73:de:40:30:e5:0a:89:dc:9d:
    f1:95:6c:ea:98:26:86:7d:1c:c1:da:ba:31:bf:24:
    69:f2:5a:ab
coefficient:
    14:89:ce:e3:69:2e:53:39:b8:63:26:cd:e7:45:9d:
    f5:7b:cd:32:5e:34:3f:98:d7:b2:be:9e:c5:f6:32:
    34:54:93:3e:8b:d0:ed:ab:51:8f:09:53:51:7c:d4:
    16:aa:8e:72:a8:ed:ea:52:d7:27:0c:bc:3e:28:91:
    e2:47:ae:57
```

- Extraire la clé publique de la clé privée et sauvegarder le résultat dans le fichier rsapubkey.pem

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl rsa -in rsakey.pem -pubout -out rsapubkey.pem
writing RSA key
```

Chiffrement de la clé RSA par l'algorithme

Nous allons maintenant utiliser l'algorithme AES256 pour chiffrer la clé privée.

- Ecrire la commande qui permet de chiffrer le fichier rsakey.pem et produit ainsi un fichier rsakeyencaes.pem.

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl enc -AES256 -in rsakey.pem -out rsakeyencaes.pem
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

┌──(root㉿kali)-[/home/fedi]
└─# openssl enc -AES256 -pbkdf2 -in rsakey.pem -out rsakeyencaes.pem
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
```

Chiffrement/déchiffrement de données avec RSA

- Ecrire la commande qui permet de chiffrer le fichier initial fichier_nom_eleve avec la clé publique rsapubkey.pem et produit ainsi un fichier fichier_nom_eleve.rsaenc (utiliser l'instruction openssl rsautl).

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl rsautl -pubin -inkey rsapubkey.pem -in fedibenamor -encrypt -out fichier_chiff_rsa

The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
```

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl pkeyutl -pubin -inkey rsapubkey.pem -in fedibenamor -encrypt -out fichier_chiff_rsa
```

- Ecrire la commande qui permet de déchiffrer le fichier fichier_nom_eleve.rsaenc et produit ainsi un fichier fichier_nom_eleve.rsadec.

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl  pkeyutl -inkey rsakey.pem -in fichier_chiff_rsa -decrypt -out fichier_dechiff_rsa
```

- Vérifier l'égalité des deux fichiers fichier_nom_eleve et fichier_dechiff_rsa.

```
┌──(root㉿kali)-[/home/fedi]
└─# cat fedibenamor
test

┌──(root㉿kali)-[/home/fedi]
└─# cat fichier_dechiff_rsa
test
```

3. SIGNATURE NUMERIQUE

Génération d'une empreinte d'un fichier

Pour signer un document on calcule d'abord une empreinte de ce document. L'instruction à utiliser pour calculer l'empreinte est :

- Calculer la valeur de l'empreinte du fichier fichier_nom_eleve avec l'algorithme MD5 et la mettre dans un fichier fichier_nom_eleve.md5.

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl dgst -md5 -out fedibenamor.md5 fedibenamor
```

- Quelle est la taille de cette empreinte ?

```
┌──(root㉿kali)-[/home/fedi]
└─# ls -l fedibenamor.md5
-rw-r--r-- 1 root root 51 Apr 21 08:40 fedibenamor.md5

┌──(root㉿kali)-[/home/fedi]
└─# cat fedibenamor.md5
MD5(fedibenamor)= d8e8fca2dc0f896fd7cb4cb0031ba249
```

- Calculer la valeur de l'empreinte du même fichier avec l'algorithme SHA1 et la mettre dans un fichier fichier_nom_eleve.sha1.

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl dgst -sha1 -out fedibenamor.sha1 fedibenamor
```

- Quelle est la taille de cette empreinte ?

```
┌──(root㉿kali)-[/home/fedi]
└─# ls -l fedibenamor.sha1
-rw-r--r-- 1 root root 60 Apr 21 08:43 fedibenamor.sha1

┌──(root㉿kali)-[/home/fedi]
└─# cat fedibenamor.sha1
SHA1(fedibenamor)= 4e1243bd22c66e76c2ba9eddc1f91394e57f9f83
```

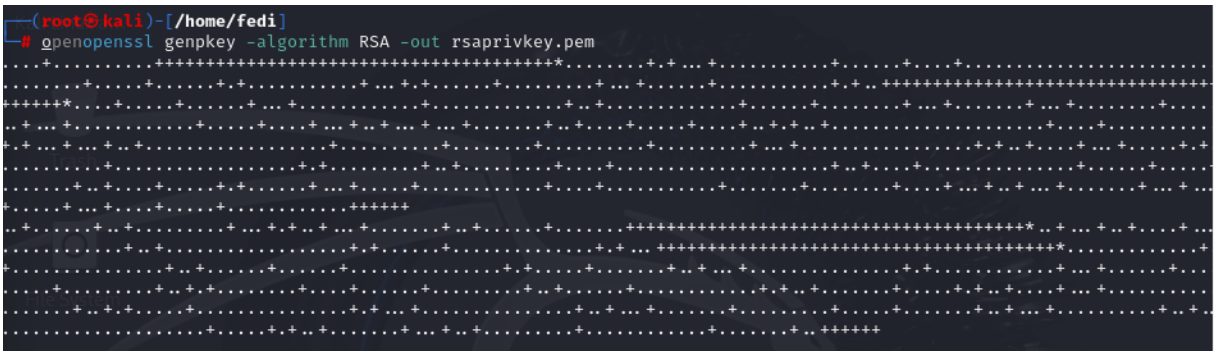
- Comparer le résultat des deux fonctions de hachage. Qu'est-ce que vous remarquez ?

Les fichiers "fedibenamor.sha1" et "fedibenamor.md5" contiennent des valeurs de hachage générées à partir de deux algorithmes de hachage différents : SHA-1 et MD5. Les longueurs des hachages sont respectivement de 60 et 51 caractères. Pour améliorer la sécurité, il est conseillé d'adopter des algorithmes de hachage plus robustes comme SHA-256 ou SHA-3.

Signature d'un fichier

- Signer un document revient à signer son empreinte. L'instruction à utiliser dans ce cas est :

```
┌──(root㉿kali)-[/home/fedi]
└─# openopenssl genpkey -algorithm RSA -out rsaprivkey.pem
```



```
┌──(root㉿kali)-[/home/fedi]
└─# openssl pkeyutl -sign -in fedibenamor.sha1 -inkey rsaprivkey.pem -out fichier_sig
```

- Il reste ensuite à vérifier que l'empreinte ainsi produite dans le fichier fichier_nom_eleve.sha1 est la même que l'on peut calculer.

```
┌──(root㉿kali)-[/home/fedi]
└─# openopenssl pkeyutl -verify -in fichier_sig -pubin -inkey rsapubkey.pem -out empreinte2

pkeyutl: No signature file specified for verify
```

4. CERTIFICAT NUMERIQUE

Génération de la clé privée

- Générer une paire de clés RSA de taille 1024 bits et stocker le résultat dans le fichier server_cle.pem

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl genrsa -out server_cle.pem 1024
```

Génération d'une requête de création d'un certificat

- Créer un fichier de demande de signature de certificat (CSR Certificate Signing Request) :

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl req -new -key server_cle.pem -out serveur_cert.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]:AU
State or Province Name (full name) [Some-State]:tunis
Locality Name (eg, city) []:tunisie
Organization Name (eg, company) [Internet Widgits Pty Ltd]:esprit
Organizational Unit Name (eg, section) []:twin
Common Name (e.g. server FQDN or YOUR name) []:fedi
Email Address []:benamor.fedi@esprit.tn

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

Signature du certificat

Afin de signer le certificat deux possibilités sont offertes :

- Auto signer le certificat
- Signer le certificat par une autorité de certification (AC)

Auto signature d'un certificat

- Signer le fichier server.cert à l'aide de la clé privée contenant dans le fichier server_cle.pem et stocker le résultat dans le fichier server_cert.crt. Le certificat doit avoir une période de validité d'un an.

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl req -new -x509 -days 365 -key server_cle.pem -out serveur_cert.crt
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]:AU
State or Province Name (full name) [Some-State]:Tunis
Locality Name (eg, city) []:tunisie
Organization Name (eg, company) [Internet Widgits Pty Ltd]:esprit
Organizational Unit Name (eg, section) []:twin
Common Name (e.g. server FQDN or YOUR name) []:fedi
Email Address []:benamor.fedi@esprit.tn
```

- Afficher le contenu du certificat en format texte

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl x509 -in serveur_cert.crt -text -noout
Certificate:
    Data:
        Version: 3 (0×2)
        Serial Number:
            5c:6e:64:fd:44:18:83:82:46:24:06:ad:6b:d6:05:d0:4e:e8:79:3a
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = AU, ST = Tunis, L = tunisie, O = esprit, OU = twin, CN = fedi, emailAddress = benamor.fedi@espri
n
        Validity
            Not Before: Apr 21 14:05:07 2024 GMT
            Not After : Apr 21 14:05:07 2025 GMT
        Subject: C = AU, ST = Tunis, L = tunisie, O = esprit, OU = twin, CN = fedi, emailAddress = benamor.fedi@espr
tn
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:b2:b1:f6:dc:46:be:68:44:91:38:56:6d:54:41:
                    2c:73:25:8e:d7:ff:2f:6a:f2:56:0f:7f:c1:c7:2e:
                    6a:df:5a:98:29:06:77:37:3d:8a:ce:b3:09:a5:48:
                    fc:ae:a5:ad:a1:ba:cb:6d:9c:7f:58:f5:cb:70:65:
                    b6:3e:7a:b0:bd:de:e4:9f:c1:46:dd:72:75:28:ce:
                    fd:78:0d:14:4a:b3:1c:99:61:77:e9:17:1e:c2:76:
                    2d:33:f7:1f:9b:3c:4f:71:5b:dd:7b:68:ee:d0:9e:
                    a2:97:b5:96:2a:8d:a0:b8:59:1e:2c:ee:c5:f5:59:
                    03:24:c2:cc:86:fc:d1:b9:bf
```

```
                Exponent: 65537 (0×10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                2F:EF:CE:C9:49:BA:CC:CE:1F:95:61:E3:36:2C:E1:2A:47:79:63:AF
            X509v3 Authority Key Identifier:
                2F:EF:CE:C9:49:BA:CC:CE:1F:95:61:E3:36:2C:E1:2A:47:79:63:AF
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        0e:a9:d4:bd:1d:90:24:07:ff:86:fb:f4:7c:ae:1d:c9:15:34:
        d3:20:53:8f:f4:d8:22:72:8c:a3:76:14:42:02:30:4a:bf:66:
        ff:aa:51:85:41:78:0e:19:25:00:6a:fa:2a:00:18:c8:82:1f:
        9d:f7:51:d4:ab:80:cf:d2:1c:60:f5:3d:57:eb:4d:2a:69:46:
        3b:19:00:e8:62:45:ce:37:24:e9:30:cc:ee:05:04:8a:00:d2:
        8e:50:60:98:1c:98:d8:ae:0d:c7:9a:3d:c7:d5:2e:b2:34:10:
        dd:6e:ae:1d:5d:c9:c8:c9:2f:14:be:5f:a4:a6:fb:68:83:c4:
        32:7e
```

Signature par une autorité de certification (AC)

- La première étape consiste à générer une clé privée RSA pour l'AC de taille 2048 bits et de stocker le résultat dans le fichier cakey.pem

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl genrsa -out cakey.pem 2048
```

- Générer un certificat pour l'AC ayant une période de validité 730 jours et stocker le résultat dans le fichier ca.crt.

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl req -new -x509 -days 730 -key cakey.pem -out ca_crt
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]:AU
State or Province Name (full name) [Some-State]:Tunis
Locality Name (eg, city) []:Tunisie
Organization Name (eg, company) [Internet Widgits Pty Ltd]:esprit
Organizational Unit Name (eg, section) []:twin
Common Name (e.g. server FQDN or YOUR name) []:fedi
Email Address []:benamor.fedi@esprit.tn
```

- Signer la demande du certificat du serveur (le fichier server.csr) par l'autorité de certification AC en utilisant l'instruction suivante :

```
┌──(root㉿kali)-[/home/fedi]
└─# openssl x509 -req -in serveur_cert.pem -out server_cert_signed.crt -CA ca_crt.crt -CAkey cakey.pem -CAcreateseri
al -CAserial ca.srl
Certificate request self-signature ok
subject=C = AU, ST = tunis, L = tunisie, O = esprit, OU = twin, CN = fedi, emailAddress = benamor.fedi@esprit.tn
Could not open file or uri for loading CA certificate from ca_crt.crt: No such file or directory
```