



**Disciplina:** INE 5680 - Segurança da Informação e de Redes

**Professora:** Carla Merkle Westphall

### Tarefa Prática – OpenSSL no Linux (www.openssl.org)

Para executar a tarefa:

- Use screenshots para documentar a execução dos comandos.
- Executar todas as questões (até 32).
- Entregar **as questões marcadas em amarelo** no moodle: entregar a **saída obtida nas questões + todos os arquivos gerados** (arquivos de teste, arquivos de chaves, etc).
- Compactar TUDO num único arquivo para entregar.
- Usar a Kali-Linux que já tem o openssl instalado, mas atualize o openssl para a última versão! Você pode usar sua própria máquina Linux para realizar a tarefa.

**\*\* Atualize a Kali-Linux com a última versão do openssl:** `apt-get install openssl`

```
root@kali:~# openssl version
OpenSSL 1.1.0h 27 Mar 2018
root@kali:~#
```

### GERAR PAR DE CHAVES RSA e entender seus componentes:

1.) a) Gerar sua chave privada usando o comando:

**`openssl genrsa -aes256 -out seunome.privada.pem 2048`**

Exemplo de comando:

```
root@kali:~/Documents/rotOpenSSL# openssl genrsa -aes256 -out carla.privada.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
Enter pass phrase for carla.privada.pem:
Verifying - Enter pass phrase for carla.privada.pem:
```

Formato PEM: "PEM format is simply base64 encoded data surrounded by header lines." ([https://www.openssl.org/docs/man1.1.0/crypto/PEM\\_read\\_bio\\_X509\\_REQ.html](https://www.openssl.org/docs/man1.1.0/crypto/PEM_read_bio_X509_REQ.html))

O arquivo `seunome.privada.pem` terá o seguinte formato de criptografia PEM (PEM ENCRYPTION FORMAT):

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: AES-256-CBC, 2F35552885460E55B5E36C8BFF96A0D1

... dados codificados em base64 ...

-----END RSA PRIVATE KEY-----
```

b) Explique o que é o parâmetro `-aes256` do comando.

c) Explique o que significa a seguinte linha do arquivo `seunome.privada.pem` ([https://www.openssl.org/docs/man1.1.0/crypto/PEM\\_read\\_bio\\_X509\\_REQ.html](https://www.openssl.org/docs/man1.1.0/crypto/PEM_read_bio_X509_REQ.html)):

```
DEK-Info: AES-256-CBC, 2F35552885460E55B5E36C8BFF96A0D1
```

d) Execute o comando seguinte e explique detalhadamente a saída obtida, que representa a estrutura e componentes da chave privada (olhe os slides da disciplina/outras referências para observar os parâmetros usados para criar a chave privada):

```
openssl rsa -text -in seunome.privada.pem
```

- 2.) **(Entregar)** Gere a chave pública a partir da chave privada com os comandos abaixo (guardar a chave pública no arquivo `seunome.publica.pem`). Explique a saída obtida em cada um dos comandos. **Guarde o arquivo gerado e envie o arquivo da sua chave pública junto nas respostas da tarefa.**

a) Explique a saída obtida no seguinte comando:

```
openssl rsa -in seunome.privada.pem -pubout -out seunome.publica.pem
```

```
Enter pass phrase for carla.privada.pem:
writing RSA key
```

```
Arquivo carla.publica.pem:
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAYR32Fi3R14eVbwME2jvn
2VixFdA3v2j1BsGEJRZ4PXhTAUILBzAgLf8U2sqC3T1CkJ+WegMKAHeeu5fqnuSB
2TpFpvyHBjHcqTRJjxdaVgwpc6Qhig7cVP4IXVL72dHKECSlrK9UCksU8lMTac44
L6g3om+5D6uV4c3MZbA/6kXq3lx00n0ThDE/Foe7n52OaYV+SoCmyQgtwwjzLmr5
Xh5FwxGMemldrrMcpsB0Eyu/Xi/+6y7bzSdwN+LW6upTXaSP5na+YFod6HefGZN
2s59M14F+Qp6e+xq5RVf7ekTaYr4bTU4Kc1PTETLXjeQ5pJBubsI6y+7k8MChjx9
lwIDAQAB
-----END PUBLIC KEY-----
```

- 3.) **(Entregar)** Digite o seguinte comando e depois abra o arquivo `seunome.publica.componentes`. Explique os componentes que constam nesse arquivo (<https://tools.ietf.org/html/rfc3447#appendix-A>). Comando:
- ```
openssl rsa -in seunome.privada.pem -out seunome.publica.componentes -text -noout
```

## ASSINATURA DIGITAL:

### 4.) (Entregar todos os itens)

a) Você deve assinar o arquivo fornecido na tarefa (msgPlana.txt). Para isso, crie o hash do arquivo msgPlana.txt e com a sua chave privada, assine o hash do arquivo:

```
openssl dgst -sha256 -sign seunome.privada.pem -out assinatura msgPlana.txt
```

b) Responda: qual o conteúdo do arquivo *assinatura* ? Essa assinatura garante quais características de segurança: integridade, autenticidade, confidencialidade?

### 5.) (Entregar) Verifique se o hash assinado está ok, isto é, compare o hash assinado com o hash do arquivo original usando o comando abaixo. Envie sua chave pública para que, durante a correção, possa ser feita a verificação da sua assinatura:

```
openssl dgst -sha256 -verify seunome publica.pem -signature assinatura msgPlana.txt
```

## CIFRAR ARQUIVO COM CHAVE SECRETA e CIFRAR A CHAVE SECRETA com o certificado de Carla (certificadoCarla.crt):

### 6.) (Entregar) Gerar uma chave secreta usando o comando (coloque o seu nome):

```
openssl rand -out chaveSecretaNomeAluno.bin -base64 128
```

### 7.) (Entregar) Cifrar o arquivo msgPlana.txt com a chave secreta criada na questão anterior:

```
openssl enc -aes-128-ctr -in msgPlana.txt -out msgCifrada -pass file:./chaveSecretaNomeAluno.bin
```

### 8.) (Entregar) Cifrar a sua chaveSecreta (chaveSecretaNomeAluno.bin) usando o meu certificado (certificado de Carla):

```
openssl rsautl -encrypt -oaep -inkey certificadoCarla.crt -certin -in chaveSecreta.bin -out chaveSecretaCifrada.enc
```

### 9.) (Entregar) Explique o que foi feito nas questões 6, 7 e 8. Explique também como será feito o processo de decifragem.

---

(Observação: vou verificar com os comandos abaixo!!)

```
openssl rsautl -decrypt -oaep -inkey carla.privada.pem -in chaveSecretaCifrada.enc -out chaveSecretaAlunoDecifrada.bin
```

```
openssl enc -aes-128-ctr -d -in msgCifrada -pass file:./chaveSecretaAlunoDecifrada.bin
```

```
root@kali:~/Documents# openssl enc -aes-128-ctr -d -in msgCifrada -pass file:./chaveSecretaAlunoDecifrada.bin
```

É preciso muita coragem e ousadia para conquistar sonhos. Não basta desejar com todas as forças do mundo, é preciso estar em movimento constante, correr atrás, persistir, perseverar. Você deve ser sua própria motivação pessoal porque você merece tudo isso que você deseja. E sim, você pode conseguir.

Trace um plano, siga-o sem desviar da rota. Comece pelas pequenas coisas: viva um dia de cada vez. Cada dia em que você der o seu melhor te deixará mais confiante em si mesmo e te tornará mais merecedor.

Livia Barros Calado

<https://www.mensagensreflexao.com.br/mensagem-de-motivacao-pessoal>

)

## GERAR UM CERTIFICADO AUTO-ASSINADO ("auto" porque é assinado com SUA própria chave privada):

- 9.) (Entregar) Para iniciar o processo de criação do SEU certificado, você deve inicialmente REQUISITAR uma assinatura no seu certificado (auto-assinado). A extensão **.csr** significa *Certificate Signing Request*. Usar o comando:

```
openssl req -new -key seunome.privada.pem -out certificado.csr
```

### Exemplo obtido na saída:

```
root@kali:~/Documents/openssl# openssl req -new -key carla.privada.pem -out certificacao.csr
Enter pass phrase for carla.privada.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]:BR
State or Province Name (full name) [Some-State]:SC
Locality Name (eg, city) []:Florianopolis
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UFSC
Organizational Unit Name (eg, section) []:SIN-2018
Common Name (e.g. server FQDN or YOUR name) []:Carla
Email Address []:carla.merkle.westphall@ufsc.br

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: (DEIXAR EM BRANCO)
An optional company name []:
root@kali:~/Documents/openssl#
```

- 10.) (Entregar) Agora o certificado X.509 AUTO-ASSINADO será efetivamente criado (assinado por você mesmo, usando a SUA chave privada), usando o comando: **openssl x509 -req -days 90 -sha512 -in certificado.csr -signkey seunome.privada.pem -out certificado.crt**

### Exemplo:

```
root@kali:~/Documents/openssl# openssl x509 -req -days 90 -sha512 -in certificacao.csr -signkey
carla.privada.pem -out certificado.crt
Signature ok
subject=C = BR, ST = SC, L = Florianopolis, O = UFSC, OU = SIN-2018, CN = Carla, emailAddress
= carla.merkle.westphall@ufsc.br
Getting Private key
Enter pass phrase for carla.privada.pem:
root@kali:~/Documents/openssl#
```

- 11.) Veja as informações do seu certificado usando o comando (copiar e colar a saída obtida):

**openssl x509 -text -in certificado.crt**

- 12.) **(Entregar)** Execute o comando seguinte e explique a saída obtida na sua execução:

**openssl s\_client -connect webmail.ufsc.br:443**

(ver no Livro Openssl Cookbook que foi compactado junto com esta tarefa)

ATIVAR SSL NO APACHE - forma simples que usa o certificado de teste  
(comandos testados no Kali Linux)

**(Entregar** -> Depois da execução dos comandos 13 até 20, entregar o **Screenshot** da carga da página <https://localhost> e também o **Screenshot do View do certificado digital**)

- 13.) Instalar o servidor apache: **apt-get install apache2**

- 14.) Ativar a configuração do virtual host com suporte a SSL: **a2ensite default-ssl**

```
root@kali:~# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@kali:~#
```

- 15.) Digite o comando para fazer o start: **service apache2 start**

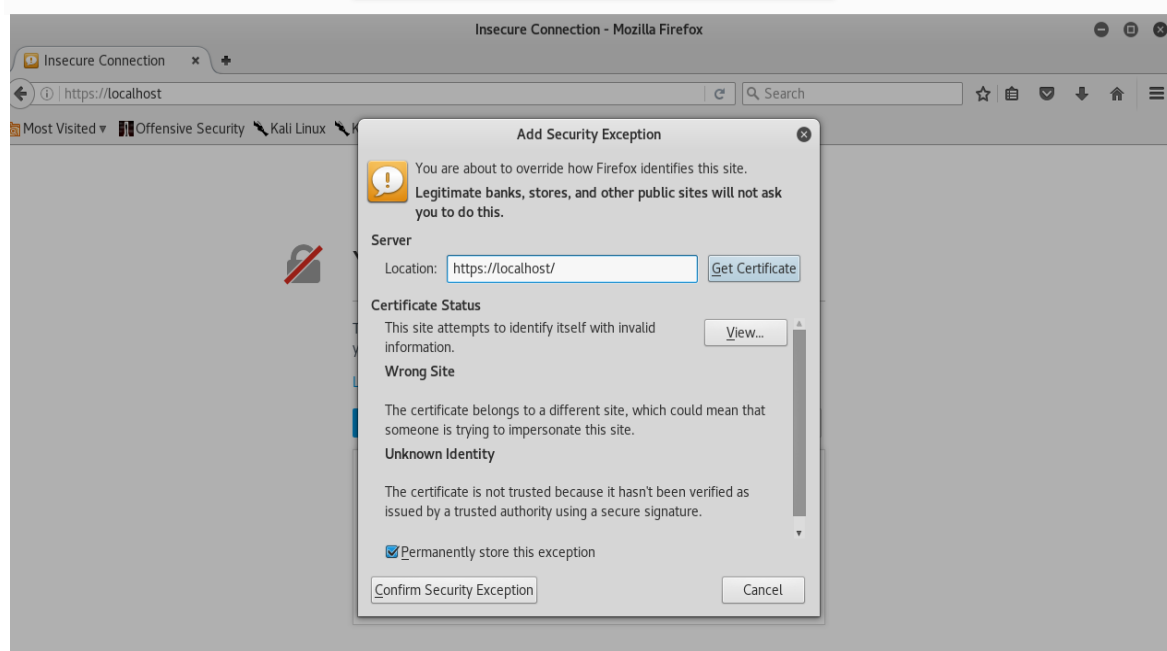
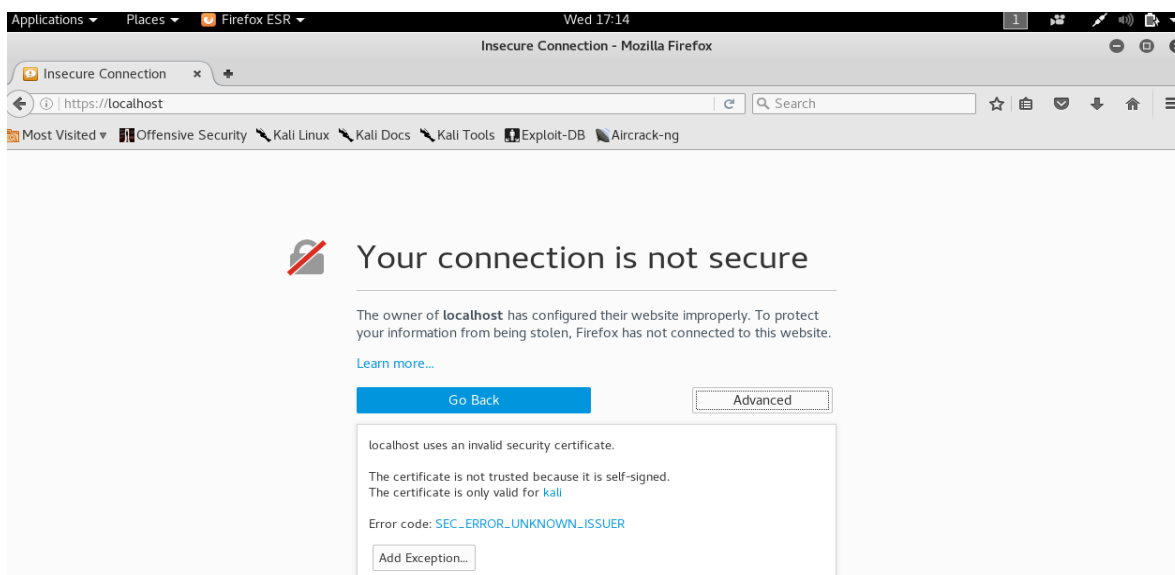
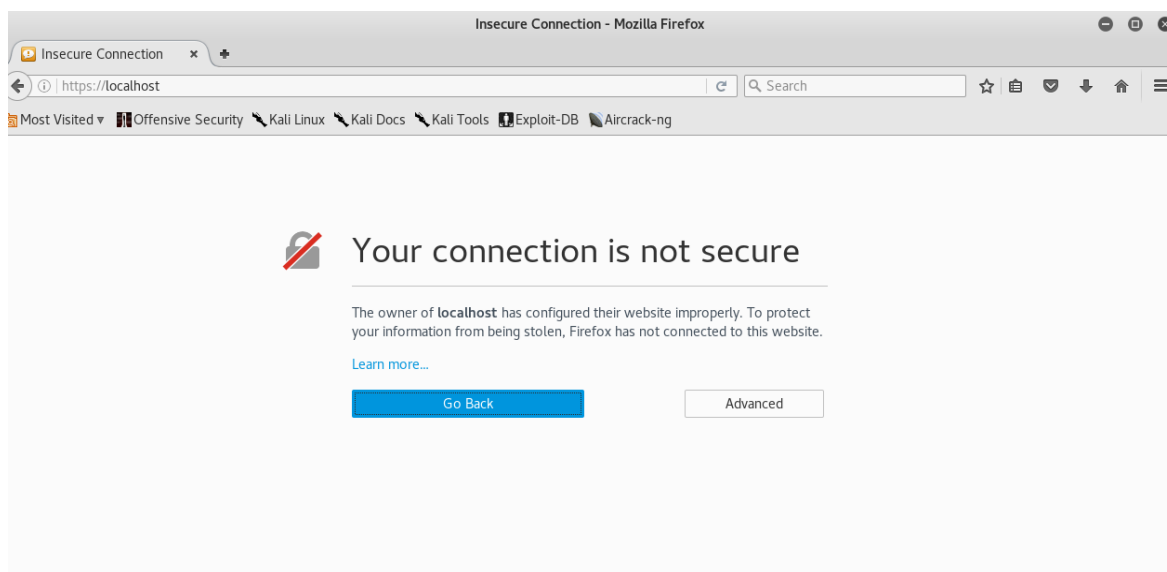
- 16.) Digite o comando para fazer o reload: **service apache2 reload**

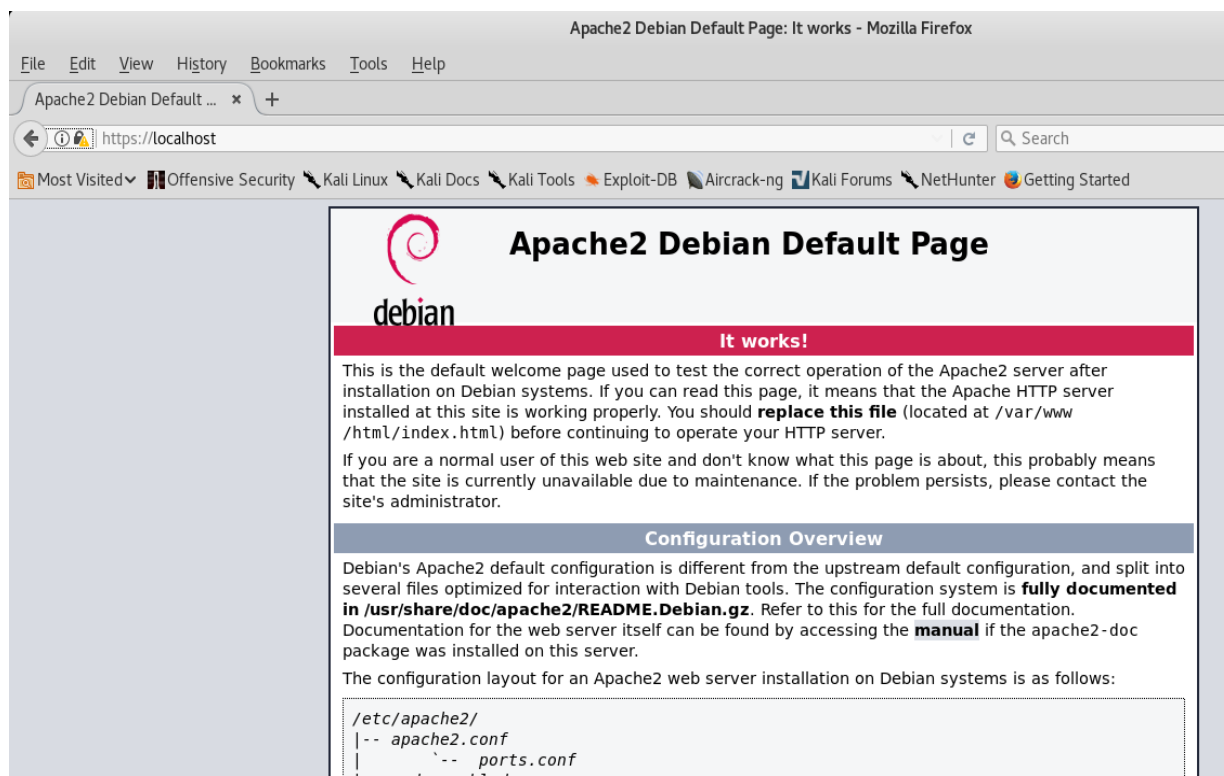
- 17.) Ativar o módulo SSL: **a2enmod ssl**

- 18.) Reiniciar o apache: **service apache2 restart**

- 19.) Verificar estado do apache: **service apache2 status**

- 20.) Acessar no browser <https://localhost>. Já deve estar funcionando. Irá aparecer na página do browser um aviso: deve-se clicar em Advanced-> Add Exception e aceitar o uso do certificado para poder ver a página de teste do apache no browser. Se você quiser ver o certificado, nesse momento, clique em View (para ver o certificado digital) e depois clique em *Confirm Security Exception*. Veja as figuras a seguir.





## ATIVAR SSL NO APACHE - criação de chave privada e certificado auto-assinado do servidor apache

(Entregar -> Depois da execução dos comandos 21 até 32, entregar o Screenshot da carga da página <https://localhost> e também e o Screenshot do View do certificado digital)

- 21.) Gerar chave privada do servidor: **openssl genrsa -aes256 -out apache.privada.pem 2048**
- 22.) Gerar requisição do certificado: **openssl req -new -key apache.privada.pem -out apache.certificado.csr**

**Atenção-** colocar "localhost" no campo **Common Name**.

**Common Name** é o nome do seu domínio. Se você tem um domínio registrado, aqui iria o nome dele, por exemplo, [www.exemplo.com.br](http://www.exemplo.com.br).

Exemplo de saída:

```
root@kali:~/Documents/openssl# openssl req -new -key apache.privada.pem -out
  apache.certificado.csr
Enter pass phrase for apache.privada.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]:BR
State or Province Name (full name) [Some-State]:SC
Locality Name (eg, city) []:Florianopolis
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UFSC
Organizational Unit Name (eg, section) []:SIN-2018
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:
```

Please enter the following 'extra' attributes  
to be sent with your certificate request

A challenge password []:

An optional company name []:

root@kali:~/Documents/openssl#

- 23.) Gerar o certificado auto-assinado: **openssl x509 -req -days 90 -sha512 -in apache.certificado.csr -signkey apache.privada.pem -out apache.certificado.crt**
- 24.) Copiar chave: **cp apache.privada.pem apache.privada.pem.copia**
- 25.) Assim, o apache não pedirá a chave a cada restart do serviço: **openssl rsa -in apache.privada.pem -out apache.privada.pem.insecure**
- 26.) Copiar chave "insegura" para a chave atual: **cp apache.privada.pem.insecure apache.privada.pem**
- 27.) Copiar chave privada para o diretório: **cp apache.privada.pem /etc/ssl/private**
- 28.) Copiar certificado para o diretório: **cp apache.certificado.crt /etc/ssl/certs**
- 29.) Mudar permissões do arquivo: **chmod 600 /etc/ssl/private/apache.privada.pem**
- 30.) Editar o arquivo default-ssl: **gedit /etc/apache2/sites-available/default-ssl.conf**

Editar as seguintes linhas colocando o caminho dos arquivos criados:

**SSLCertificateFile /etc/ssl/certs/apache.certificado.crt**

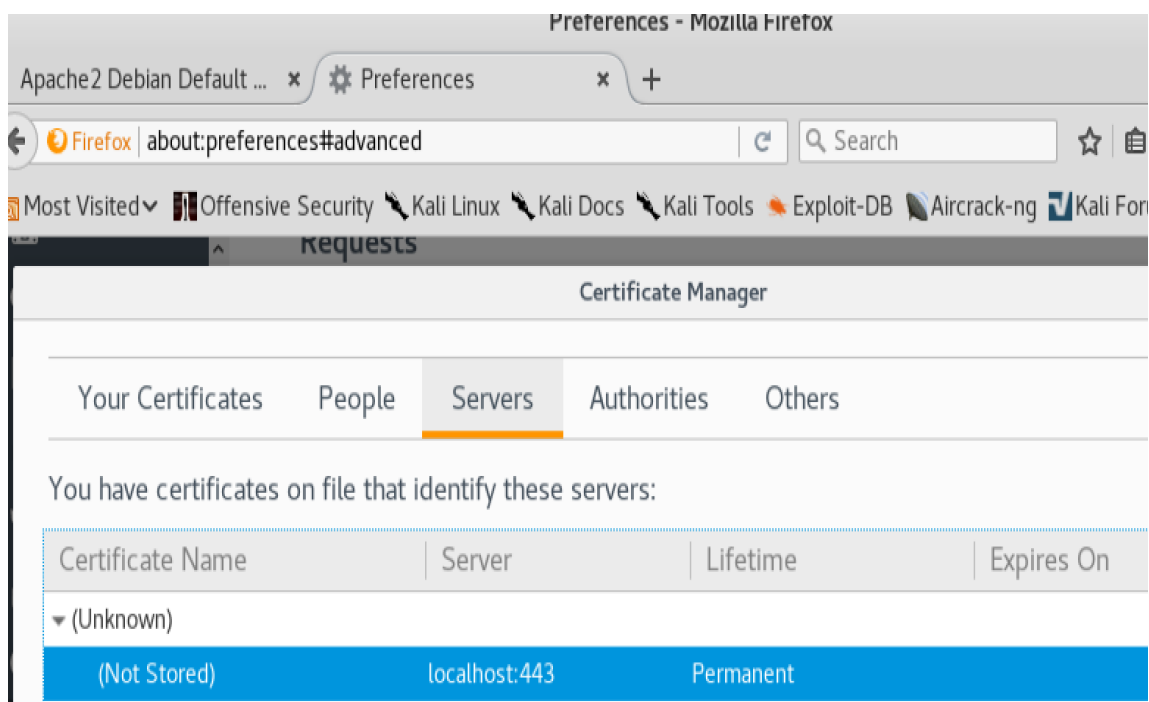
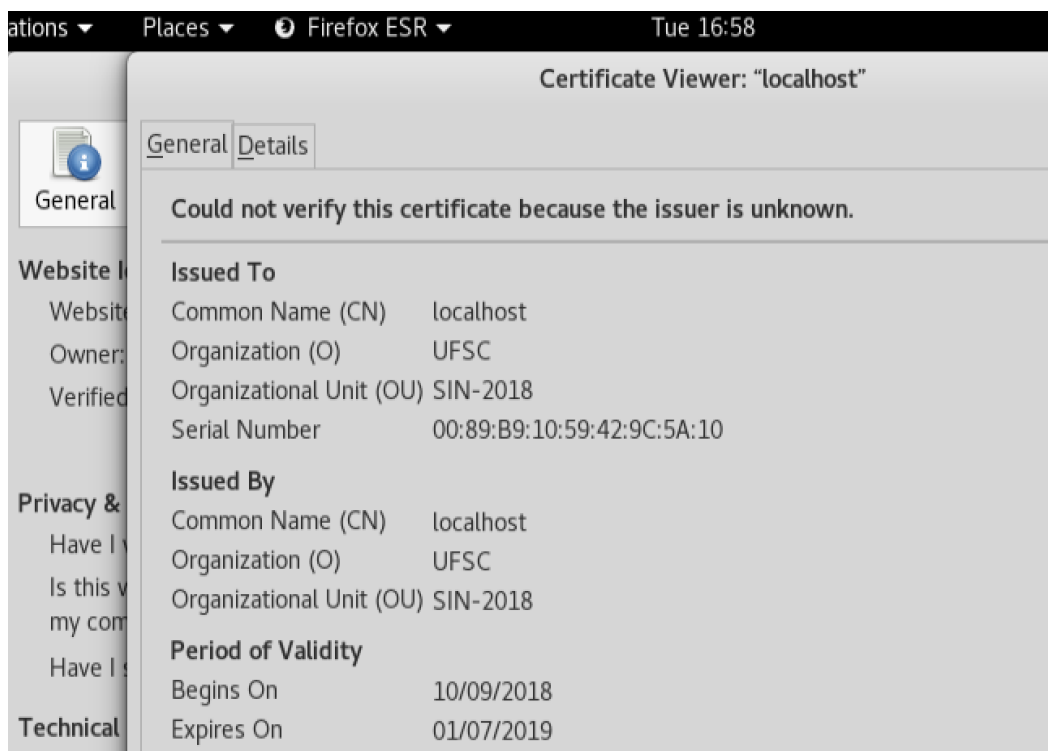
**SSLCertificateKeyFile /etc/ssl/private/apache.privada.pem**

- 31.) Reiniciar o apache (usar o apache HTTP Server Control Interface - apache2ctl) :  
**apache2ctl restart**

**Obs: O sistema vai reclamar pois é localhost (127.0.0.1) e não é nome de domínio real!**

- 32.) Acessar no browser <https://localhost> (execute um Reload na página). Já deve estar funcionando, com o certificado criado nas questões 21 em diante. Mostre o conteúdo do certificado usado (use a opção View).





### Questão opcional - Usar o Let's Encrypt para ATIVAR SSL NO APACHE2

Nessa questão, deve ser usado o Let's Encrypt (<https://letsencrypt.org/>) para ativar o SSL de algum servidor. Todos os passos desenvolvidos devem ser documentados na forma de relatório.

Você deve pesquisar e essa questão deve ser apresentada, caso seja feita.

**\*\* Detalhes de comandos:**

```
$ openssl version -a      - mostra a versão do openssl
$ openssl list-standard-commands
$ openssl list-cipher-commands
$ openssl list-message-digest-commands
$ openssl list-cipher-algorithms
$ openssl list-public-key-algorithms
$ openssl speed           - oferece um benchmark completo
$ openssl speed rsa       - testa somente o algoritmo rsa
$ openssl dgst -h
$ openssl enc -h
```

**Referências**

1. Comandos: [http://wiki.openssl.org/index.php/Command\\_Line\\_Uutilities](http://wiki.openssl.org/index.php/Command_Line_Uutilities)
2. Livro Openssl Cookbook: <https://www.feistyduck.com/library/openssl-cookbook/online/>
3. Manpages: <https://www.openssl.org/docs/manpages.html>
4. Comandos: <https://www.openssl.org/docs/man1.1.0/apps/>
5. CA própria - <https://jamielinux.com/docs/openssl-certificate-authority/index.html>
6. Simple Introduction: <https://sandilands.info/sgordon/simple-introduction-to-using-openssl-on-command-line>
7. Encrypt and decrypt files to public keys via the OpenSSL Command Line: [https://raymii.org/s/tutorials/Encrypt\\_and\\_decrypt\\_files\\_to\\_public\\_keys\\_via\\_the\\_OpenSSL\\_Command\\_Line.html#Get\\_the\\_public\\_key](https://raymii.org/s/tutorials/Encrypt_and_decrypt_files_to_public_keys_via_the_OpenSSL_Command_Line.html#Get_the_public_key)