

PKI Lab report by Anthony Coppolecchia

The screenshot shows a dual-monitor setup. The left monitor displays a web browser window titled "PKI Lab" with the URL psu.instructure.com/courses/2326423/assignment.... The right monitor displays another web browser window titled "Practice Labs | Lab" with the URL practice-labs.com/app/platform/lab.aspx. Both monitors show a terminal window titled "plabadmin@PLABUBUNTU" running on a Linux desktop. The terminal window contains several lines of command-line text, including OpenSSL commands and nano editor sessions for configuration files.

Step 2

Let's see the content of the `server.key` file using the following command:

```
openssl rsa -in server.key -text
```

Step 3

Now we will generate a certificate signing request for our client's website using the following command:

```
openssl req -new -key server.key -out server.csr -config openssl.cnf
```

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	
Task 1: Creating necessary directories	Correctly created the necessary directories		10 pts

Step 1

Copy the `openssl.cnf` file into the PKI directory using the following command:

```
cp /usr/lib/ssl/openssl.cnf ./
```

Step 2

Next, change the `openssl.cnf` file to allow for the new CA to issue certificates for any country, state or organization. Use nano to edit the file. Find the CA policy section of the file and change from "match" to "supplied". Save the file when exiting with <ctrl-X> in nano.

```
nano openssl.cnf
```

For the CA policy

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	
Task 1: Creating necessary directories	Correctly created the necessary directories		10 pts

In this screenshot I changed the directory to desktop then created a PKI directory and changed the directory to desktop/pki. Then I created a new directory called demoCA and changed the directory to it. Then I created three folders called certs, crl and newcerts. Then I edited serial to put the number 1000 in it and I showed the contents of serial. I then created a new file called index.txt and saved it to the demo ca directory I made. Then switched back to the pki folder I created. I then copied the openssl.cnf file into the pki directory I made.

PKI Lab report by Anthony Coppolecchia

The screenshot shows a dual-monitor setup. The left monitor displays a web browser window for 'PKI Lab' on 'psu.instructure.com'. It shows a step-by-step guide for generating a self-signed certificate. Step 2 involves viewing the content of the 'server.key' file using the command 'openssl rsa -in server.key -text'. Step 3 involves generating a certificate signing request (CSR) using 'openssl req -new -key server.key -out server.csr -config openssl.cnf'. A grade summary table indicates 10 pts for creating a certificate authority and 0 pts for correctly creating necessary directories, totaling 10 pts.

The right monitor displays a terminal window titled 'PLABUBUNTU' on 'practice-labs.com'. The terminal shows the output of the OpenSSL commands. It includes a long string of RSA private key data starting with '-----BEGIN RSA PRIVATE KEY-----' and ending with '-----END RSA PRIVATE KEY-----'. The terminal also shows the creation of directory structures and files for a demo CA.

In this ss i changed openssl.cnf to allow for demo ca to issue certificates and i used nano command to edit the fileand changed match to supplied . I then created a self signed certificate.

PKI Lab report by Anthony Coppolecchia

The screenshot shows a macOS desktop environment. On the left, there's a vertical dock with icons for various applications like Mail, Calendar, and Finder. Two browser windows are open: one titled "PKI Lab" showing a course assignment page on psu.instructure.com, and another titled "Practice Labs | Lab" showing a terminal session on practice-labs.com. The terminal window is running on a distribution called "PLABUBUNTU". It displays a command-line interface with several lines of text output, including file paths and command history. A table at the bottom of the desktop shows task completion status:

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	
Task 1: Creating necessary directories	Correctly created the necessary directories		10 pts

In the following ss i showed the contents of more ca.key

This screenshot continues from the previous one, showing the same desktop setup. The terminal window on the right now shows the contents of a file named "ca.key" with the command "more ca.key". The file contains a long string of encrypted private key data. The table at the bottom remains the same:

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	
Task 1: Creating necessary directories	Correctly created the necessary directories		10 pts

PKI Lab report by Anthony Coppolecchia

The screenshot shows a dual-monitor setup. The left monitor displays a Chrome browser window titled 'PKI Lab' with the URL psu.instructure.com/courses/2326423/assignment.... A sidebar on the left lists various course materials. The right monitor shows a terminal window titled 'Practice Labs | Lab' with the URL practice-labs.com/app/platform/lab.aspx. The terminal session is running on a PLABUBUNTU system. The terminal window title bar says 'aci [PRACTICELABS]' and the window title is 'PLABUBUNTU'. The terminal displays several commands related to RSA key generation and content of server.key files.

PKI Lab Assignment Content:

Step 2
Let's see the content of the `server.key` file using the following command:
`openssl rsa -in server.key -text`

Step 3
Now we will generate a certificate signing request for our [client's website](#) using the following command:
`openssl req -new -key server.key -out server.csr -config openssl.cnf`

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	
Task 1: Creating necessary directories	Correctly created the necessary directories		10 pts

Terminal Session (plabadmin@PLABUBUNTU: ~/Desktop/PKI\$):

```
BIGGosCC6rRHAQKAQEAA3sPURJRF90zQD00m0V2H-KaH60T9Y6djes702o7wv0
v$LR699jsEr4n9ff05zsaJ1bwSbRyeI1j2xcRoFcn107e5ctBB8YHr3gVUJ7
600sqJm+<+IaVkn5dtrRZ+Hn1o94vtzJhJUNh50TKccxve4EfJvnk8sIUqau837
lRvn3j>/W+yiscfNNUKEZIEkrCB3XwPBuVol929n4x43DPB1V27t7zhTRNE
00AcydsxRfwfJ+ZCVnOy2xG1D5pbpxX/3zZYLAHKTY6XDNHH191BD3mqDBP14
m+mttPYR10z2f2H0EBubhg2v1MSnfvdxnw9xugC0KCAQ+B+Ls+F+jBwXhLssid4
D1L2LhSqwShuhrsAvaseOpzQ0Py+EW05yyTSGrTrc610x+HqVfxOrgc1VgzzCc
KJa80Lkar0icmVLvz/X01j3juw+n1NgxjCA/y1wP/PQRBkoq2xQx3fBmh4IJ
exIE0HsNEAcnwEh01pegfDmhB1o2GTdnTQT4kYLaa9yHsew8TRDmhslr
Iys1gcvtkuzufeAqftxeP4ahzx1b5gFvLHt58s2ctu0dLjArC49nqr+1q6
vQeqJHjVtj3+UnCBeMy0w102k0HmHmlt/pzdZmcShNyE3SRk6f925noxQnm6
kESBa1BAQDr0hVvVSp9uV5QMTWHevwGSt0Y0d6/jzeK3n0Cbgp8d1vPMmNz
h4t1u5Zq189l60pAqg16z1775L19501166100978948589100000000000000000
MJOEThR0080wB971bn398eUtr1L0Co2zwz/C1zFaahs0LE515n-/9bhig-A83
AO2Jk/Z1g5UEWEHHA/0P5AVKLVx+1Kge3i4/1090+jezP1LY6kdVlLxQ1FDvHB
AKN91YgDme7LsV24TeeLvoQHMsK1XpZyhp1wYZncFkCVSnEN3.22hbH008cg
Tzadr1V5+rWrwE4ag7JhceksCbx8Zao1B48c1fxG2Q+24a8Tr0f0tvk40p67a
+OB2RcrNAvQ1VpdH0qzLbzpzbhbt/L0Nf7hYu442ad52QmLs5mGkE4f3t/2Fpg
OTEkgCZU2Ltgch12xF35OnqdXBuF9N0wBPFK404t2vdAdYo+A752SGzCuettX
225kjupMzkJ19UsPcmXFX3R/E5JSOUFdfrCqWhLkXAmDs/4mGB6+KqbVhLME4q
7u7Ew3fMT5jrtdzGaJqvYLUT147n6a000uzCoSOC/uEg3Pw1nEuJ7UBGSEIwg
IZo+5dmeEU97Bn+yLDLfbd0en4xVkfHzvR0R7ySp4z0B0yQ80tvyvlk=
-----END RSA PRIVATE KEY-----
```

PKI Lab Assignment Content (continued):

Step 2
Next, we will generate a 4096-bit RSA public and private key pair for our website's certificate. The private key will be encrypted using AES 256.
`openssl genrsa -aes256 -out server.key 4096`

**server.key is where the keys will be stored.
Enter pass phrase for server.key: **customer** (not be displayed)

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	
Task 1: Creating necessary directories	Correctly created the necessary directories		10 pts

Terminal Session (plabadmin@PLABUBUNTU: ~/Desktop/PKI\$):

```
f09xxnlBYCYcl0BhtW7TaRf)CaKnSAU1JW/J1MhVyoFnSwqdxVhosNxU7+eCz41h
02kepB8nxt4t4bsX/lfyCw==  
----- ENCRYPTED PRIVATE KEY -----  
plabadmin@PLABUBUNTU:~/Desktop/PKI$ ls -la
total 32
drwxr-x--x 3 plabadmin plabadmin 4096 May 30 10:38 .
drwxr-x--x 3 plabadmin plabadmin 4096 May 30 10:32 ..
-rw-rw-r-- 1 plabadmin plabadmin 1493 May 30 10:38 ca.crt
drwxrwxr-x 5 plabadmin plabadmin 4096 May 30 10:34 demoCA
-rw-r--r-- 1 plabadmin plabadmin 1854 May 30 10:37 ca.key
plabadmin@PLABUBUNTU:~/Desktop/PKI$ openssl denrsa -aes256 -out server.key 4096
Invalid command 'denrsa'; type "help" for a list.
plabadmin@PLABUBUNTU:~/Desktop/PKI$ openssl denrsa -aes256 -out server.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....  
.
.
.
e: 15 d5537 (0x010001)  
Enter pass phrase for server.key:  
Verifying - Enter pass phrase for server.key:  
plabadmin@PLABUBUNTU:~/Desktop/PKI$
```

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop with three windows open:

- PKI Lab** (psu.instructure.com/courses/2326423/assignment...): A course assignment page for Exercise 1. It asks for a challenge password ("seed") and displays an optional company name ("Website certificate for my client 1").
- Practice Labs | Lab** (practice-labs.com/app/platform/lab.aspx): A practice lab interface for aci [PRACTICELABS]. It shows a terminal window with the command "openssl req -new -key server.key -out server.csr -config openssl.cnf" being run. The terminal output shows the generation of an RSA private key and the creation of a certificate signing request (CSR).
- PLABUBUNTU**: A terminal window on a Linux desktop environment. It shows the user "plabadmin" running the same command as the Practice Labs terminal. The terminal output is identical, showing the generated RSA private key and CSR.

Step 2
Let's see the content of the `server.key` file using the following command:
`openssl rsa -in server.key -text`

Step 3
Now we will generate a certificate signing request for our [client's website](#) using the following command:
`openssl req -new -key server.key -out server.csr -config openssl.cnf`

Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks	0 pts No Marks	10 pts
---	----------------------	----------------------	--------

PKI Lab report by Anthony Coppolecchia

Step 2

Let's see the content of the `server.key` file using the following command:

```
openssl rsa -in server.key -text
```

Step 3

Now we will generate a certificate signing request for our [client's website](#) using the following command:

```
openssl req -new -key server.key -out server.csr -config openssl.cnf
```

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	10 pts
Task 1: Creating necessary directories			

Step 4

Now we will have our CA issue a certificate for the website using the CSR we just created by using the following command:

```
openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf
```

Enter pass phrase for ca.key: **IAMCA1.** (used before for creating the CA)

Select Yes to sign the certificate.

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks	10 pts
Task 1: Creating necessary directories			

PLABUBUNTU

Activities Terminal May 30 10:41

```
plabadmin@PLABUBUNTU: ~/Desktop/PKI
plabadmin@PLABUBUNTU: ~/Desktop/PKI
-----END RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----
```

PLABUBUNTU

Activities Terminal May 30 10:44

```
plabadmin@PLABUBUNTU: ~/Desktop/PKI
plabadmin@PLABUBUNTU: ~/Desktop/PKI
Serial Number: 4096 (0x1000)
Validity
    Not Before: May 30 14:44:44 2024 GMT
    Not After : May 30 14:44:44 2025 GMT
Subject:
    countryName          = US
    stateOrProvinceName = New Jersey
    organizationName     = customer1
    organizationalUnitName= Unit1
    commonName           = website.com
    emailAddress         = company1@gmail.com
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    Netscape Comment:
        OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
        56:24:AB:58:6A:E1:24:16:7C:55:90:45:4F:A5:E3:09:AB:2D:FF
    X509v3 Authority Key Identifier:
        keyId:c9:98:A3:0A:50:82:BD:E2:EB:BB:14:81:30:C6:0
6
Certificate is to be certified until May 30 14:44:44 2025 GMT (365 days)
Sign the certificate? [y/n]:
```

PKI Lab report by Anthony Coppolecchia

The screenshot displays a dual-monitor setup. The left monitor shows the PKI Lab interface from psu.instructure.com, while the right monitor shows a terminal session on aci [PRACTICELABS] PLABUBUNTU.

PKI Lab (Left Monitor):

- Step 2:** A terminal window shows the command: `openssl rsa -in server.key -text`.
- Step 3:** A terminal window shows the command: `openssl req -new -key server.key -out server.csr -config openssl.cnf`.
- A grade table:

	Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks	0 pts No Marks	10 pts
--	---	----------------------	----------------------	--------

Terminal Session (Right Monitor):

- The terminal shows the output of the RSA private key creation command, which includes a long string of characters and ends with "-----END RSA PRIVATE KEY-----".
- The terminal shows the creation of a certificate signing request (CSR) using the command: `openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf`.
- The terminal shows the user entering a pass phrase for ca.key: `IAMCA1.` (used before for creating the CA).
- The terminal shows the user selecting `Yes` to sign the certificate.
- A grade table:

	Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks	0 pts No Marks	10 pts
--	---	----------------------	----------------------	--------

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop environment with two browser windows and a terminal window on a Linux desktop.

Top Left Browser: PKI Lab - psu.instructure.com/courses/2326423/assignment...
Content: Step 2
Let's see the content of the server.key file using the following command:
openssl rsa -in server.key -text

Top Right Browser: Practice Labs | Lab - practice-labs.com/app/platform/lab.aspx
Content: aci [PRACTICELABS] PLABUBUNTU Activities Terminal May 30 10:41 plabadmin

Terminal Window: plabadmin@PLABUBUNTU: ~/Desktop/PKI
Content: A long RSA private key output starting with "-----BEGIN RSA PRIVATE KEY-----".

Bottom Left Browser: PKI Lab - psu.instructure.com/courses/2326423/assignment...
Content: Task 1 – Setting up the web browser
Now that we have created our certificate files, we must tell our web browser to accept certificates signed by our fake CA.
Step 1
We need to add website.com to our hosts file. This will cause the web browser to be directed to the local machine instead of the internet. Use nano to edit the hosts file in the /etc directory. Admin permissions are required, so we must use the sudo (Super User Do) command.

Bottom Right Terminal: plabadmin@PLABUBUNTU: ~/Desktop/PKI
Content: nano /etc/hosts
Output: A hosts file entry for "website.com" added to the end of the file.

Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks Correctly created the necessary directories	0 pts No Marks	10 pts
---	---	----------------------	--------

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop environment with three main windows:

- PKI Lab** (Chrome): A course assignment page from psu.instructure.com.
- Practice Labs | Lab** (Chrome): A PracticeLabs session titled "PLABUBUNTU".
- Terminal**: An open terminal window showing a long command-line session for generating an RSA private key.

Step 2
Let's see the content of the `server.key` file using the following command:
`openssl rsa -in server.key -text`

Step 3
Now we will generate a certificate signing request for our client's website using the following command:
`openssl req -new -key server.key -out server.csr -config openssl.cnf`

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories	Correctly created the necessary directories	10 pts

Step 2
Press Control + X to exit, and select Yes to save the changes, and then press enter to return to the terminal

Step 3
Modern web browsers use DNS over HTTPS (DoH) to locate the IP Address of websites. We must override this behavior to force Firefox to use the local resolver (hosts file).
Choose preference, search for network settings
Launch Firefox and navigate to this setting. Options > General > Scroll down to Network Settings > Uncheck the Enable DNS over HTTPS box. Close Firefox for the changes to take place. If you do not see this setting, then you can move on to the next step.

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories	Correctly created the necessary directories	

PKI Lab report by Anthony Coppolecchia

The screenshot shows a dual-monitor setup. The left monitor displays a web browser window for 'PKI Lab' on 'psu.instructure.com'. It shows a step-by-step guide for generating a certificate signing request (CSR). Step 2 involves running the command 'openssl rsa -in server.key -text'. Step 3 involves generating a CSR with the command 'openssl req -new -key server.key -out server.csr -config openssl.cnf'. A table below tracks progress: Exercise 1 (Creating a certificate authority) is at 10 pts (Full Marks), and Task 1 (Creating necessary directories) is at 0 pts (No Marks). The right monitor displays a terminal window on 'PLABUBUNTU' showing the output of the CSR generation command. The output includes a long string of characters representing the CSR and ends with '-----END RSA PRIVATE KEY-----'. The terminal also shows the creation of a certificate using OpenSSL's self-signing command.

Step 2
Let's see the content of the `server.key` file using the following command:
`openssl rsa -in server.key -text`

Step 3
Now we will generate a certificate signing request for our [client's website](#) using the following command:
`openssl req -new -key server.key -out server.csr -config openssl.cnf`

Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks Correctly created the necessary directories	0 pts No Marks	10 pts
---	--	----------------------	--------

plabadmin@PLABUBUNTU: ~/Desktop/PKI

```
BIGGosC6rRHAQKAQEAA3sPURJRF90zQD00oV2zKaH60t9Y6djes702o7wv0  
v$LRG99jsEr4n9ff05zsa1jbwSsRyeI1j2xRoFcu107e5ctB8YHr3gVUJ7  
600sqJm+wcIAVkn5dr4RZtW9t94vtzJhJUNh50TKccxve4EfVnk8sIUqkU837  
lRvn3jW+yiscfNXXBUKEZ1cER8CD3xWPBuVol929n4x43DPB1V27y7zhIRNE  
00AcysdxsRxwfJ+ZCVnOy2gLDspbxX/3vZYLAHKTY6XDNNN191803dmpDBP14  
m+nMtPYR10z2f2H0EBubgh2v1MSnfvdNx9xugCOKCAQB+Ls+fJ+bmXKhssid4  
D1L2L1hSquShuhsRaBavseOp2Q0y+EW05yyTSGPTrc610x+HqVfxOrgcVgzzCc  
Kjat8jLkar0icmVLzX/V013jyuw+nGNgjCA/y1v/p/PQRBKOg2xQ03f8mMh4lJ  
exIE0HSNEJACmwEJhdh111pegfDmhB1ozGTdNTQT4KYLaa9yHsese8WTRDmhsxlr  
Iys1gcvf1kuz2feAqftxeP4ahzx1b5gFvLHt58s2ctu0dJarc499qr+1q6  
v1eqfJhV1jC3+UnCBMeY0v102kOHmhl/ptDzmcShNyE3SRK6F925noxQnm6  
kEsBa1BAQDRohhVVPsp9fQzQTMHEhvG50t0Yd6/jezK3n0Cbgp8d1pMMNz  
6t4kXfJhV1jC3+UnCBMeY0v102kOHmhl/ptDzmcShNyE3SRK6F925noxQnm6  
jJOEThR0808mB971bn3948eUtr1OCo2zWz/C1bFaahSL151n//9b1egA83  
AO2jUK/Z115UEWEHHA/C0PSAVKLvA+4gep3i4j109o+jezP1Ly6kdvlvLxQ1FDvHb  
AKNq1YgDme+zLsv24GTeeLvoQHWSK1XpZyhp1wYZncFkCVSnEN122HbH08cg  
Tzadr1V5+rWlwR4ogJ7hxceksCgbXZao1B4Bc1fxG20+24a8VTrfotVtyk40p67a  
+OB2RcrNAv0lvpdhz1bpozbhbt/L0nf7hyUu442ab52qnl5sawke4Ef3t/2Fpg  
OTEkgCu21tGch12xF35OnqdXBfu9N0w8PFK40tIzVdAdYo+A752SGzQeUTtX  
225kjupMzkj19UsPcmXfX3R/2E5JS0UfdfrsCq4hLkXAmDs/4mGB6+kqbVhLME4q  
7u7Ew3fMT5jrtdzGaJqgvLUT1476a00UzCo5OC/Ueg3P81nEuJu7UBGEIwg  
IZo+AsdneEU9Bn+yLD1Bfdden4xVHFZtvR0R2ySp4z0B0yQ80tvyvlk=
```

Step 4
Now we will test the certificate in `server.pem` by launching an OpenSSL TLS server
`openssl s_server -cert server.pem -www`

(Pass phrase: **customer**)

Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks Correctly created the necessary directories	0 pts No Marks	1
---	--	----------------------	---

plabadmin@PLABUBUNTU: ~/Desktop/PKI

```
OpenSSL Generated Certificate  
X509v3 Subject Key Identifier:  
56:24:A8:5B:6:EA:24:C9:16:7C:55:90:45:4F:A5:E3:09:AB:2D:FF  
X509v3 Authority Key Identifier:  
keyid:C9:9B:8B:A3:56:0A:76:09:3F:50:82:BD:E2:EB:BB:14:81:30:C6:0  
6  
Certificate is to be certified until May 30 14:44:44 2025 GMT (365 days)  
Sign the certificate? [y/n]:  
1 out of 1 certificate requests certified, commit? [y/n]:  
Write out database with 1 new entries  
Database Updated  
plabadmin@PLABUBUNTU: ~/Desktop/PKI$ sudo nano /etc/hosts  
[sudo] password for plabadmin:  
plabadmin@PLABUBUNTU: ~/Desktop/PKI$ cd ~/Desktop/PKI  
bash: cd: /Desktop/PKI: No such file or directory  
plabadmin@PLABUBUNTU: ~/Desktop/PKI$ cat server.key server.crt >server.pem  
plabadmin@PLABUBUNTU: ~/Desktop/PKI$ openssl s_server -cert server.pem  
Enter pass phrase for server.pem:  
Using default temp DH parameters  
ACCEPT
```

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop with three windows open:

- PKI Lab**: A browser window showing a course assignment page on psu.instructure.com.
- Practice Labs | Lab**: A browser window showing a PLABUBUNTU terminal session on practice-labs.com.
- PLABUBUNTU**: A terminal window showing a user named plabadmin logged in. The terminal displays a long string of RSA private key data.

Step 2
Let's see the content of the server.key file using the following command:
`openssl rsa -in server.key -text`

Step 3
Now we will generate a certificate signing request for our client's website using the following command:
`openssl req -new -key server.key -out server.csr -config openssl.cnf`

Exercise 1: Creating a certificate authority Task 1: Creating	10 pts Full Marks Correctly created the necessary directories	0 pts No Marks	10 pts
--	---	-------------------	--------

Step 5
Now we will go into Firefox and manually add the certificate. Open Firefox and go to the following link:
<https://website.com:443>
Note: This will take a few minutes to load the page. You will likely get a warning about the certificate. Be able to explain what that warning means. Bypass the warning message.
In the network setting, choose advanced options and choose "don't use the proxy"

Exercise 1: Creating a certificate authority Task 1: Creating necessary directories	10 pts Full Marks Correctly created the necessary directories	0 pts No Marks	
--	---	-------------------	--

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop with three windows open:

- PKI Lab**: A Chrome window displaying the URL psu.instructure.com/courses/2326423/assignment.... It contains a step-by-step guide for creating a certificate authority.
- Practice Labs | Lab**: A Chrome window displaying the URL practice-labs.com/app/platform/lab.aspx. It shows a terminal session on a "PLABUBUNTU" system with the user "plabadmin". The terminal output is a long string of RSA private key data.
- PLABUBUNTU**: A terminal window showing the user "plabadmin" at the prompt "plabadmin@PLABUBUNTU: ~/Desktop/PKI". The terminal displays the same RSA private key data as the Practice Labs window.

Step 2
Let's see the content of the `server.key` file using the following command:
`openssl rsa -in server.key -text`

Step 3
Now we will generate a certificate signing request for our [client's website](#) using the following command:
`openssl req -new -key server.key -out server.csr -config openssl.cnf`

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories	Correctly created the necessary directories	

Step 5
Now we will go into Firefox and manually add the certificate. Open Firefox and go to the following link:
<https://website.com:4433>

Note. This will take a few minutes to load the page. You will likely get a warning about the certificate. Be able to explain what that warning means. Bypass the warning message.
In the network setting, choose advanced options and choose "don't use the proxy"

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories	Correctly created the necessary directories	

PKI Lab report by Anthony Coppolecchia

The screenshot displays two monitors. The left monitor shows a Mac OS X desktop with a Chrome browser window titled "PKI Lab" open to psu.instructure.com/courses/2326423/assignment... . The right monitor shows a Linux desktop with a Firefox browser window titled "Practice Labs | Lab" open to practice-labs.com/app/platform/lab.aspx . Both screens show terminal windows and file explorers.

Step 2
Let's see the content of the server.key file using the following command:

```
openssl rsa -in server.key -text
```

Step 3
Now we will generate a certificate signing request for our client's website using the following command:

```
openssl req -new -key server.key -out server.csr -config openssl.cnf
```

Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks Correctly created the necessary directories	0 pts No Marks	10 pts
---	---	----------------------	--------

Step 7
Choose import, then select our certificate and press open. Check the box with Trust this CA to identify Websites and press okay.

The screenshot shows the Firefox Certificate Manager dialog box. It asks if you want to trust "fakeca.com" for identifying websites. The "Trust this CA to identify websites" checkbox is checked. Below it, a note says: "Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available)." There are "View" and "Examine CA certificate" buttons, and "Cancel" and "OK" buttons at the bottom.

PKI Lab report by Anthony Coppolecchia

Step 2
Let's see the content of the `server.key` file using the following command:

```
openssl rsa -in server.key -text
```

Step 3
Now we will generate a certificate signing request for our [client's website](#) using the following command:

```
openssl req -new -key server.key -out server.csr -config openssl.cnf
```

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories	Correctly created the necessary directories	10 pts

Step 8
Now go back and reload <https://website.com:443/>. Notice the difference in how the SSL is handled. Notice the lock icon has changed on the URL line. You may now close the Firefox browser.

Step 9
Once back in the terminal, press **Control + C** to break free from the terminal session. This is the "server" in the OpenSSL Suite that has allowed us to test the SSL certificates. In the next part of the exercise, we will configure the Apache web server with the TLS certificate for our site.

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories	Correctly created the necessary directories	

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop with two browser windows and a terminal window on a Linux desktop.

Left Browser Window: psu.instructure.com/courses/2326423/assignment... (PKI Lab)

Right Browser Window: practice-labs.com/app/platform/lab.aspx (aci [PRACTICELABS])

Terminal Window: plabadmin@PLABUBUNTU: ~/Desktop/PKI

Task 2: Let's see the content of the server.key file using the following command:

```
openssl rsa -in server.key -text
```

Task 3: Now we will generate a certificate signing request for our client's website using the following command:

```
openssl req -new -key server.key -out server.csr -config openssl.cnf
```

Table:

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating necessary directories	10 pts	

Task 8: Now go back and reload <https://website.com:443/>. Notice the difference in how the SSL is handled. Notice the lock icon has changed on the URL line. You may now close the Firefox browser.

Task 9: Once back in the terminal, press Control + C to break free from the terminal session. This is the "s_server" in the OpenSSL Suite that has allowed us to test the SSL certificates. In the next part of the exercise, we will configure the Apache web server with the TLS certificate for our site.

Table:

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating necessary directories	10 pts	

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

```
plabadmin@PLABUBUNTU: ~/Desktop/PKI
```

```
56:24:AB:5B:6E:EA:24:C9:16:7C:55:9D:45:4F:A5:A3:E3:09:AB:2D:FF
X509v3 Authority Key Identifier:
keyId:C9:9B:8B:A3:56:0A:76:A9:3F:50:82:BD:E2:EB:BB:14:81:C6:0
6
Certificate is to be certified until May 30 14:44:44 2025 GMT (365 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
plabadmin@PLABUBUNTU:~/Desktop/PKI$ sudo nano /etc/hosts
[sudo] password for plabadmin:
plabadmin@PLABUBUNTU:~/Desktop/PKI$ cd /Desktop/PKI
bash: cd: /Desktop/PKI: No such file or directory
plabadmin@PLABUBUNTU:~/Desktop/PKI$ cat server.key server.crt >server.pem
plabadmin@PLABUBUNTU:~/Desktop/PKI$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
1401469905601344:error:14094118:SSL routines:ssl3_read_bytes:tlsv1 alert unknown
ca:../ssl/record/rec_layer_s3.c:1543:SSL alert number 48
```

PKI Lab report by Anthony Coppolecchia

The screenshot shows a dual-monitor setup. The left monitor displays a web browser window for 'PKI Lab' on 'psu.instructure.com' and a terminal window titled 'PKI'. The right monitor displays a web browser window for 'Practice Labs | Lab' on 'practice-labs.com' and a terminal window titled 'PLABUBUNTU'.

Step 2: Let's see the content of the server.key file using the following command:

```
openssl rsa -in server.key -text
```

Step 3: Now we will generate a certificate signing request for our client's website using the following command:

```
openssl req -new -key server.key -out server.csr -config openssl.cnf
```

Exercise 1:

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating necessary directories		10 pts

Step 4: Go back to the other terminal window and ensure both the CERT and KEY files are in the /etc/apache2/ssl directory utilizing the following command:

```
ls -la /etc/apache2/ssl
```

Step 5: Go back to the apache2 directory and switch to the sites-available directory using the following command:

```
cd /etc/apache2
cd sites-available
```

Exercise 1:

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating necessary directories		

PKI Lab report by Anthony Coppolecchia

The screenshot shows a dual-monitor setup. The left monitor displays a Chrome browser window for 'PKI Lab' with the URL psu.instructure.com/courses/2326423/assignment.... The right monitor displays a 'Practice Labs | Lab' session titled 'aci [PRACTICELABS]' on a 'PLABUBUNTU' desktop environment.

Step 2: Let's see the content of the server.key file using the following command:
openssl rsa -in server.key -text

Step 3: Now we will generate a certificate signing request for our client's website using the following command:
openssl req -new -key server.key -out server.csr -config openssl.cnf

Exercise 1: Creating a certificate authority
Task 1: Creating necessary directories

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating necessary directories		10 pts

Step 6: We will create a new configuration file called website.com.conf for our website. Use the default-ssl.conf file as an example if you have problems.
sudo nano website.com.conf

Enter the following lines into the file:

```
plabadmin@PLABUBUNTU:~$ sudo nano website.com.conf
plabadmin@PLABUBUNTU:~$ cat website.com.conf
# Configuration for website.com
ServerName website.com
ServerAdmin webmaster@website.com
DocumentRoot /var/www/website.com/html
ErrorLog /var/www/website.com/logs/error.log
CustomLog /var/www/website.com/logs/access.log combined
SSLEngine on
SSLCertificateFile /etc/apache2/certs/cert.pem
SSLCertificateKeyFile /etc/apache2/certs/key.pem
```

Exercise 1: Creating a certificate authority
Task 1: Creating necessary directories

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating necessary directories		

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop with three windows open:

- PKI Lab**: A browser window showing a course assignment page on psu.instructure.com.
- Practice Labs | Lab**: A browser window showing the aci [PRACTICELABS] platform.
- PLABUBUNTU**: A terminal window running on a Linux system (Ubuntu).

Step 2
Let's see the content of the `server.key` file using the following command:
`openssl rsa -in server.key -text`

Step 3
Now we will generate a certificate signing request for our client's website using the following command:
`openssl req -new -key server.key -out server.csr -config openssl.cnf`

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating	Correctly created the necessary directories	

Step 6
We will create a new configuration file called `website.com.conf` for our website. Use the default-ssl.conf file as an example if you have problems.
`sudo nano website.com.conf`

Enter the following lines into the file:

```
<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerName website.com
        DocumentRoot /var/www/website.com/
        DirectoryIndex index.html
        SSLEngine On
        SSLCertificateFile /etc/apache2/ssl/CERT.pem
        SSLCertificateKeyFile /etc/apache2/ssl/KEY.pem
    </VirtualHost>
```

Exercise 1:	10 pts	0 pts
Creating a certificate authority	Full Marks	No Marks
Task 1: Creating necessary directories	Correctly created the necessary directories	

The terminal window shows the contents of the `website.com.conf` file being edited in nano, and the output of the `cat /etc/apache2/sites-available/website.com.conf` command.

PKI Lab report by Anthony Coppolecchia

The screenshot displays two monitors. The left monitor shows a Chrome browser window for 'PKI Lab' on 'psu.instructure.com'. It contains a 'Step 2' section with the command 'openssl rsa -in server.key -text' highlighted in yellow. Below it is a 'Step 3' section with the command 'openssl req -new -key server.key -out server.csr -config openssl.cnf' highlighted in yellow. A table below these sections tracks progress: Exercise 1 (Creating a certificate authority) is at 10 pts Full Marks, Task 1 (Creating necessary directories) is at 0 pts No Marks, totaling 10 pts. The right monitor shows a 'Practice Labs | Lab' window on 'practice-labs.com/app/platform/lab.aspx'. It features a terminal window titled 'PLABUBUNTU' with a long RSA private key output. The terminal interface includes various keyboard shortcuts at the bottom.

Exercise 1: Creating a certificate authorit Task 1: Creating necessary directories	10 pts Full Marks	0 pts No Marks	10 pts
---	----------------------	----------------------	--------

PKI Lab report by Anthony Coppolecchia

The screenshot shows a Mac desktop with three windows:

- PKI Lab**: A browser window showing a course assignment page on psu.instructure.com.
- Practice Labs | Lab**: A browser window showing the aci [PRACTICELABS] interface.
- Terminal**: A terminal window titled "plabadmin" on "PLABUBUNTU". It displays a long command-line session for generating an RSA private key, including various file operations and a base64 encoded key.

A table at the bottom summarizes the task completion status:

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories		10 pts

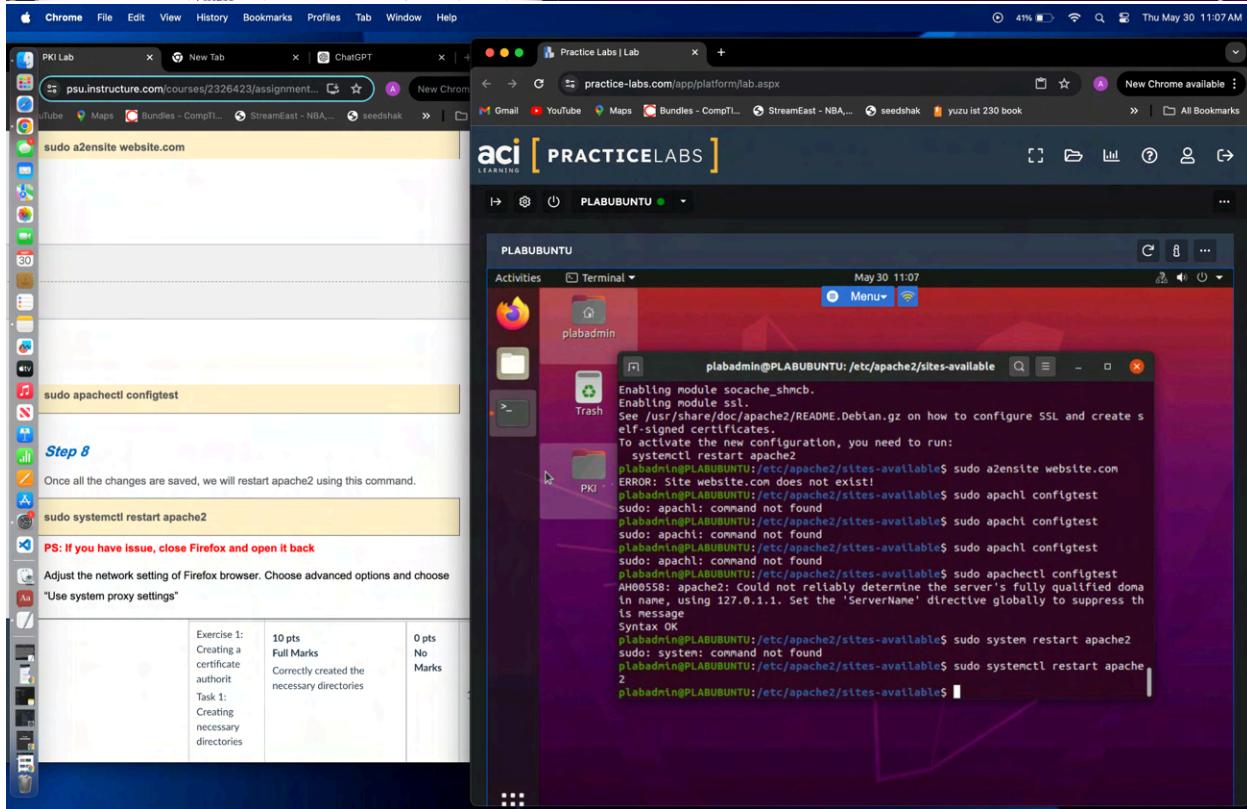
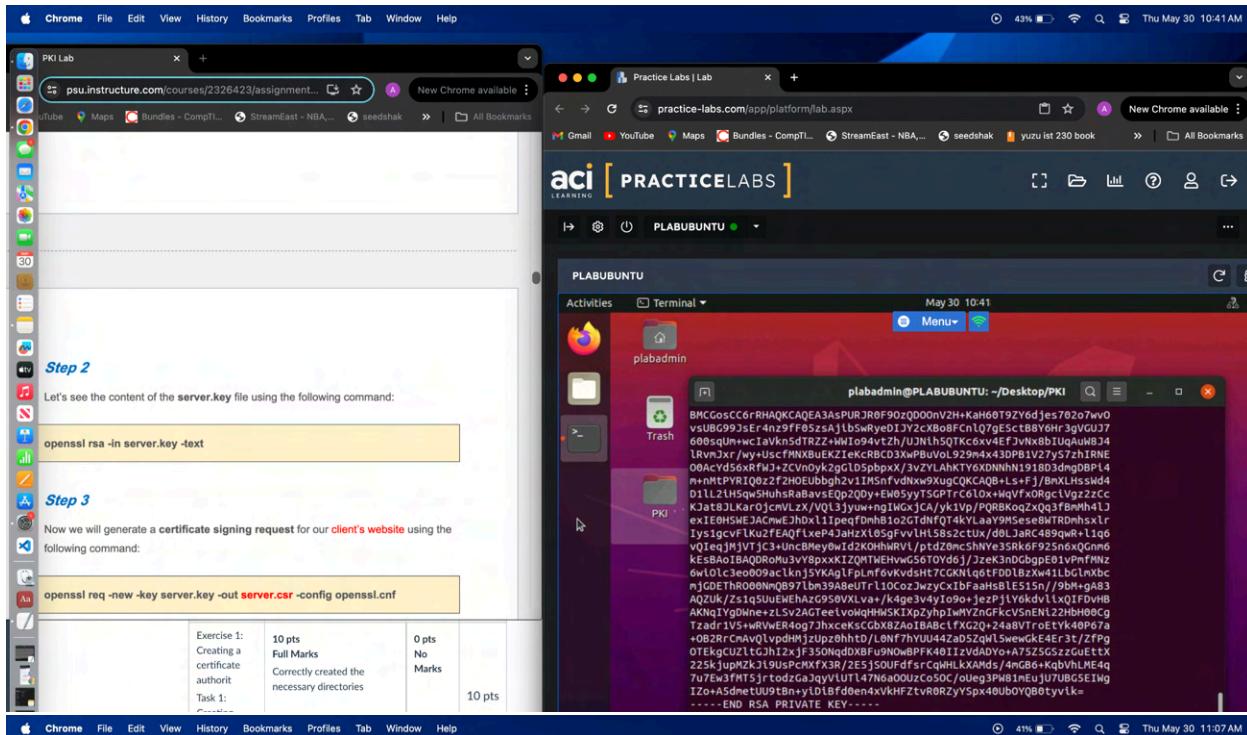
The screenshot shows a Mac desktop with three windows:

- PKI Lab**: A browser window showing a course assignment page on psu.instructure.com.
- Practice Labs | Lab**: A browser window showing the aci [PRACTICELABS] interface.
- Terminal**: A terminal window titled "plabadmin" on "PLABUBUNTU". It displays a command-line session for editing an Apache configuration file (index.html) using nano editor. The file contains HTML code for a secure website.

A table at the bottom summarizes the task completion status:

Exercise 1: Creating a certificate authorit	10 pts Full Marks	0 pts No Marks
Task 1: Creating necessary directories		

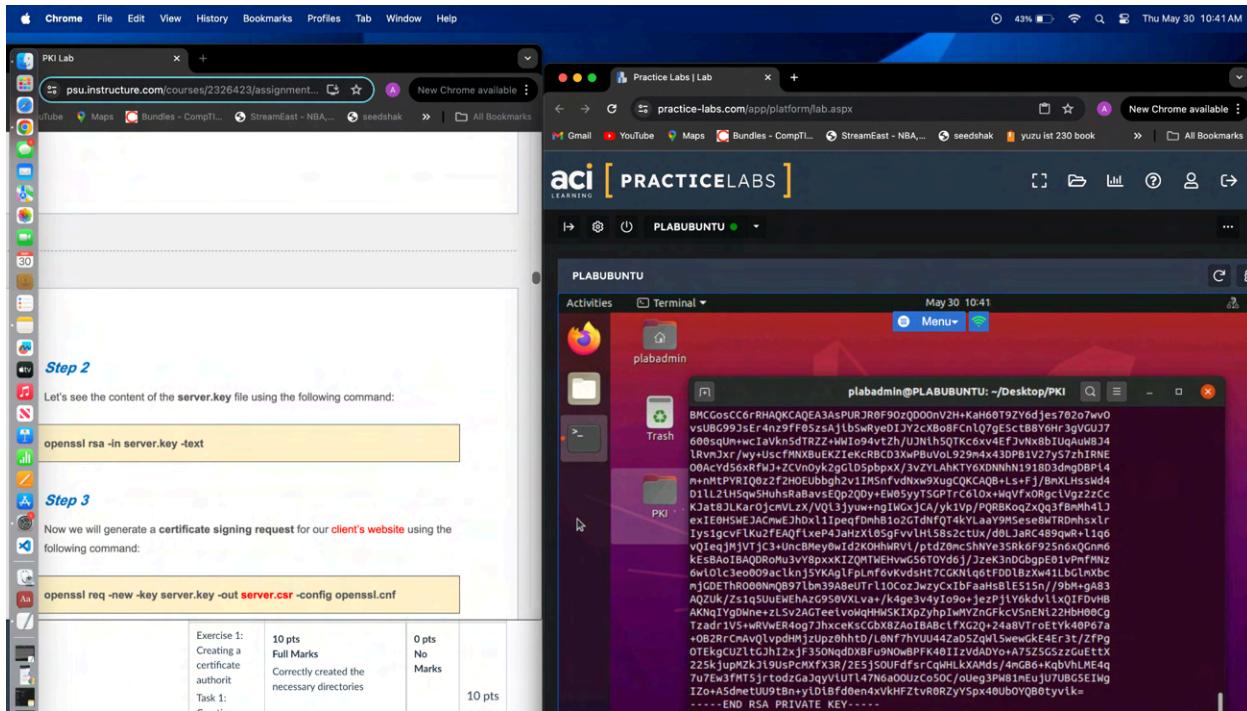
PKI Lab report by Anthony Coppolecchia



It was too hard to get my comments and text in between each screenshot so i will type my walkthrough here, next i checkek nontents of pki directory using `ls -la`. Next we generated a 4096 bit rsa public and private key pair for `website.com` and encrypted it to `-aes256`.

Next i viewed and showed the content of `server.key` file. Then i generated a certificate request signing for the `website.com`. Next we had the CA issue a certificate for the website using the

PKI Lab report by Anthony Coppolecchia



Step 2

Let's see the content of the `server.key` file using the following command:

```
openssl rsa -in server.key -text
```

Step 3

Now we will generate a certificate signing request for our [client's website](#) using the following command:

```
openssl req -new -key server.key -out server.csr -config openssl.cnf
```

Exercise 1: Creating a certificate authorit Task 1: Create	10 pts Full Marks Correctly created the necessary directo ries	0 pts No Marks	10 pts

CSR we created. Next we added website.com to our hosts file to view it in web. Next we had to override network preferences and proxy settings. Next we added the certificate to the webserver with the commands `cd ~/Desktop/PKI`

`cat server.key server.crt > server.pem`

Next we tested the cert in `server.pem` launching a `ssl tls` server then we went into firefox and manually added the certificate and bypassed the warning message net we imported our certificate into firefox next we deployed the certificate to a more permanent web server. We created a directory to store certificates in apache 2 then we created and moved the `cert.pem` and `key.pem` to the `ssl apache 2` directory then we switched to the `sites-available` directory then we created a new config file called `website.com.conf` then we created a simple wepface face using html code in the terminal.then we enabled the `ssl` module and our site config test and tested the files to make sure we get a syntax ok message. Then we restarted apache 2 and adjusted firefox network settings and reopened <https://website.com/> to make sure it is still working and active.