12th Project

Lecturer: Dr.Ayman Adel Abdelhamid

T.A: Abdelrhman Solyman

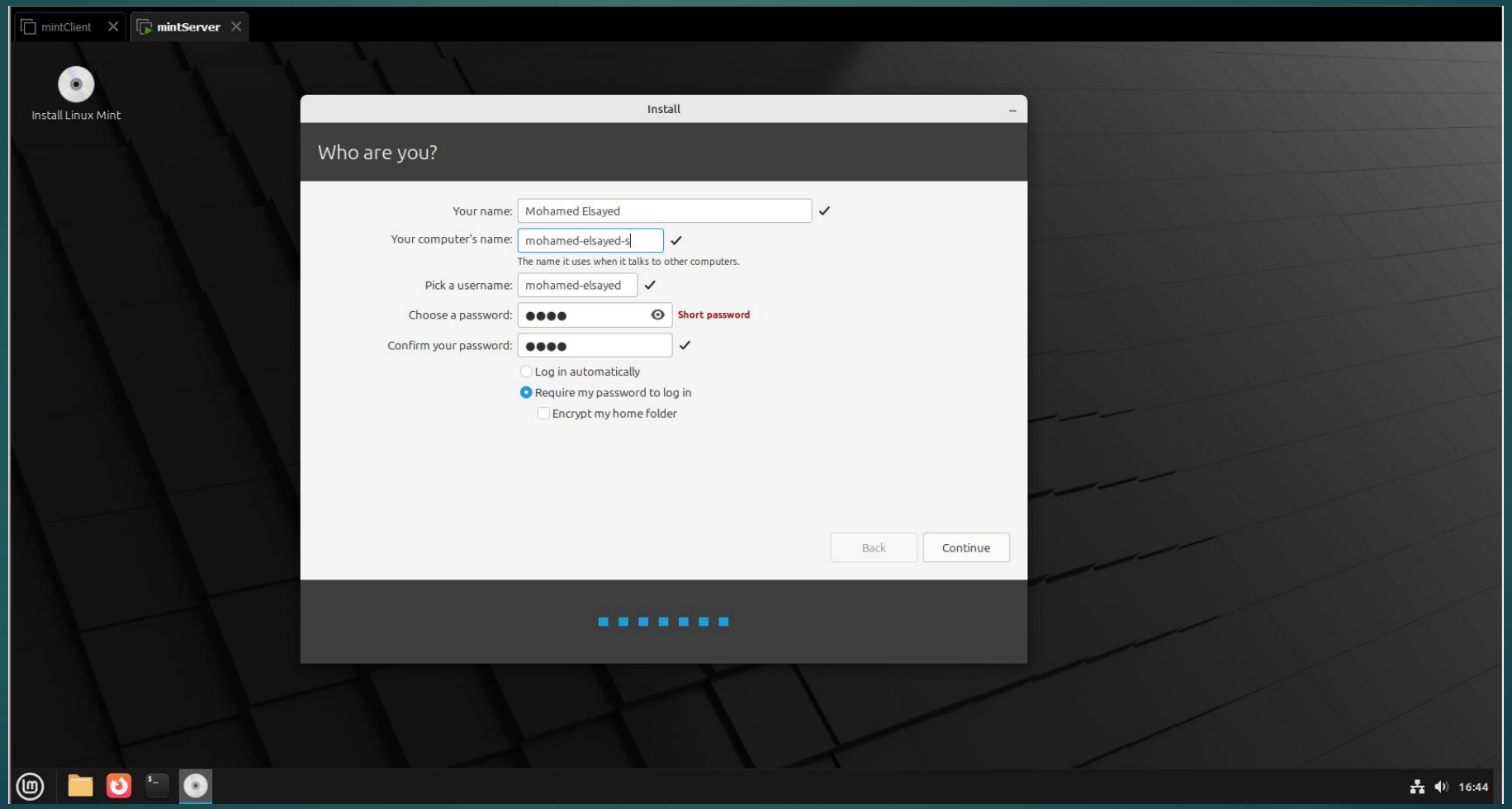
NAME:

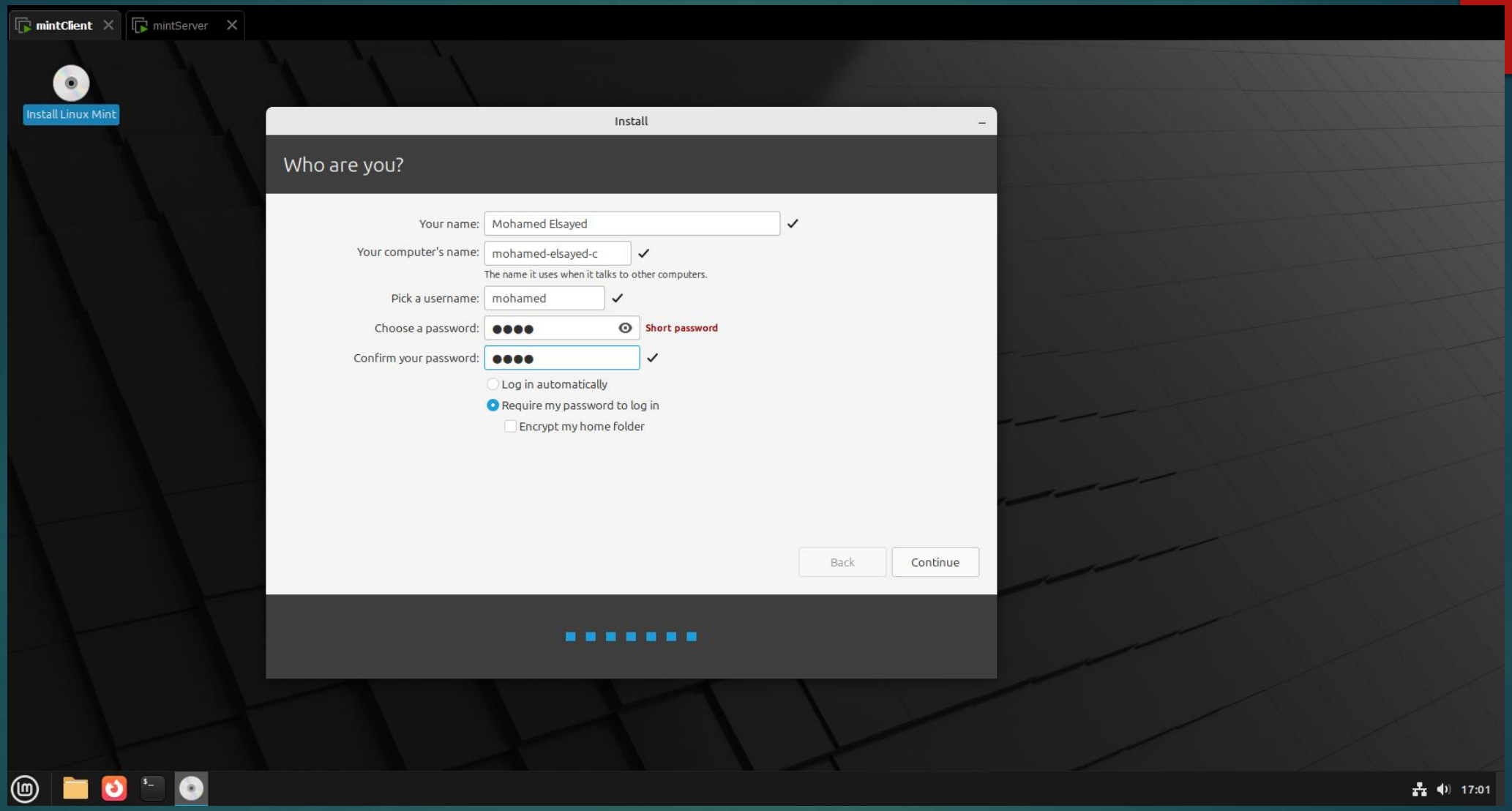
MOHAMED ELSAYED MOHAMED 221010750

OMAR SHERIF HOSNY 221010339

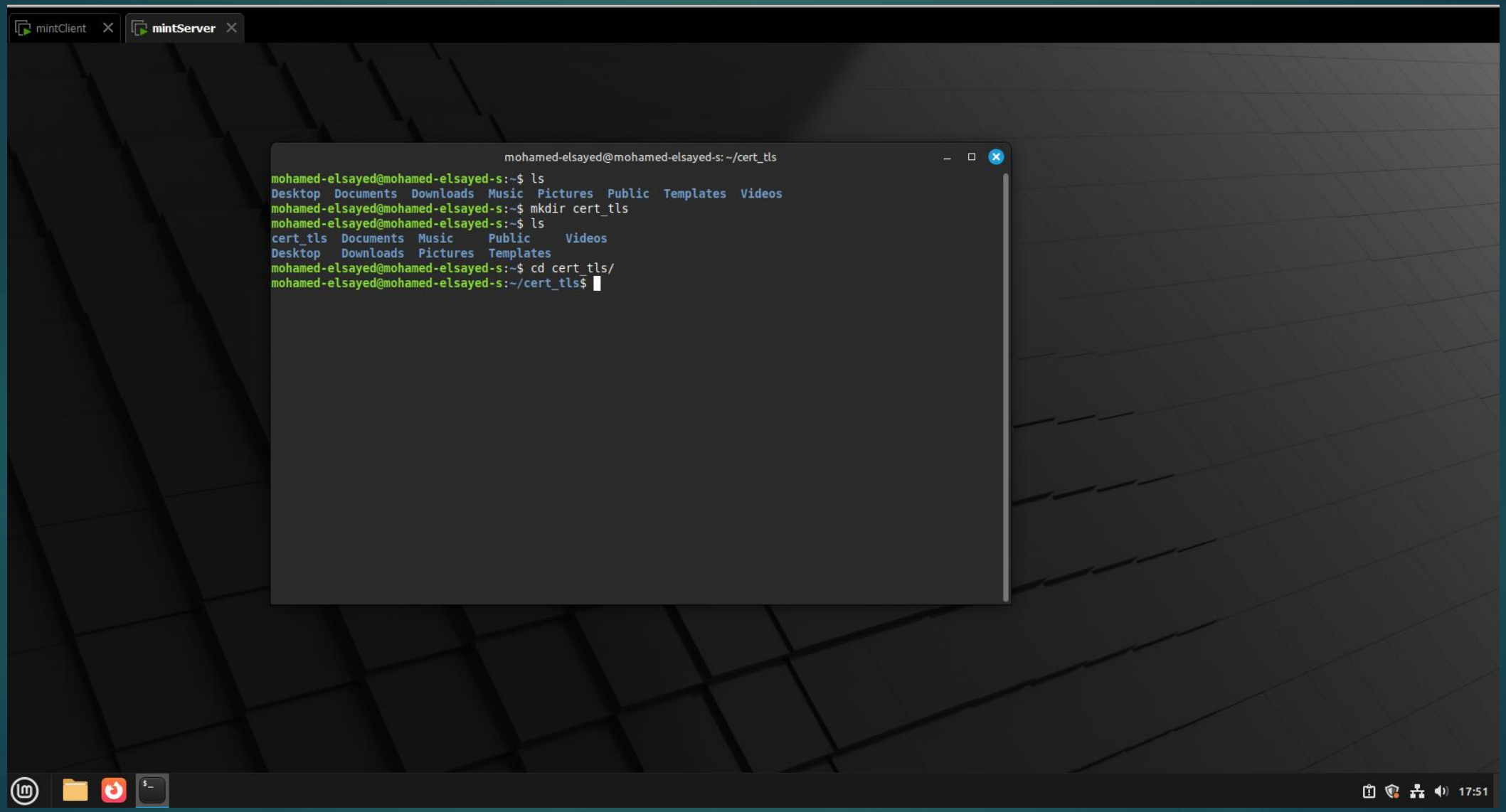
Part 1 (TLS) :

Making Mint Server and Mint client

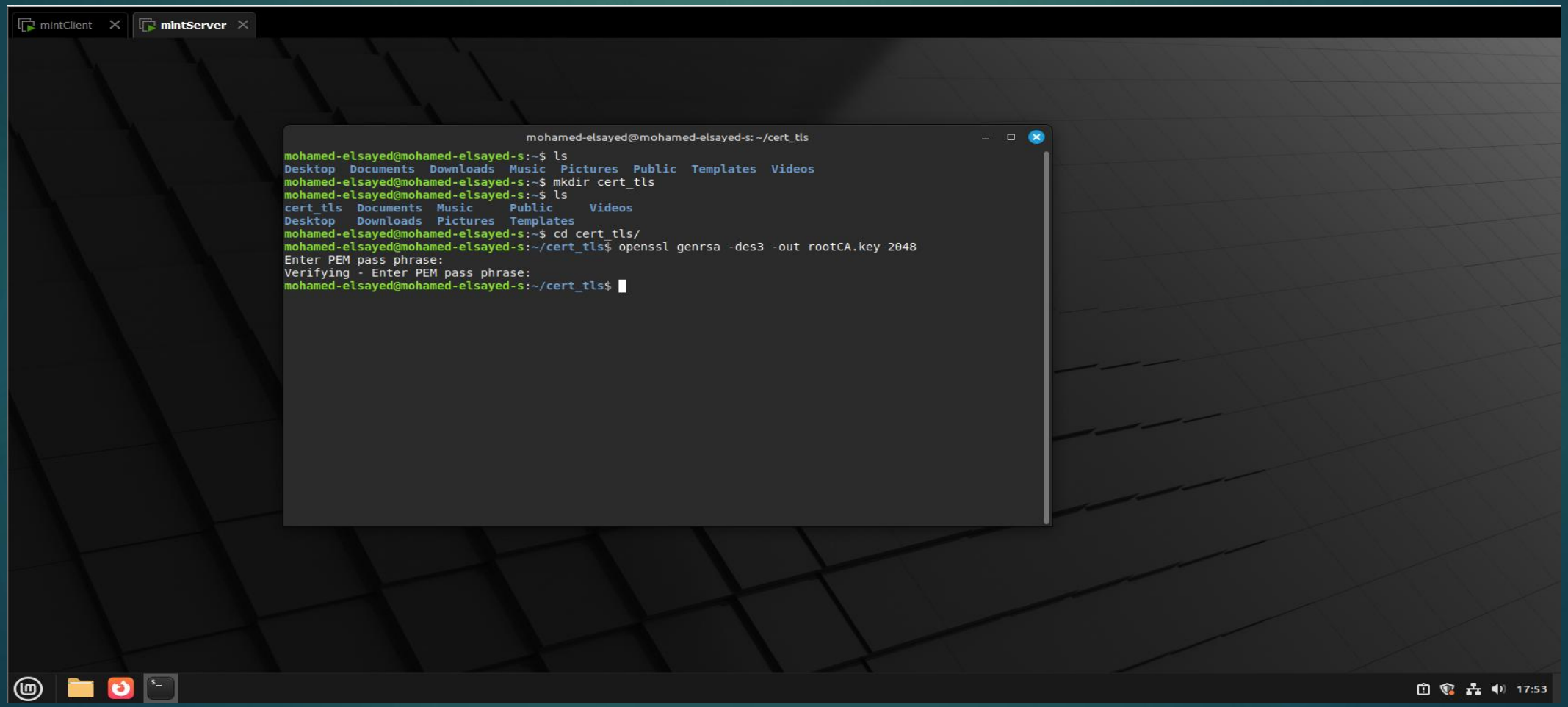



mintClient
mintServer
Install Linux Mint
Install
Who are you?
Your name:
Mohamed Elsayed
Your computer's name:
mohamed-elsayed-c
The name it uses when it talks to other computers.
Pick a username:
mohamed
Choose a password:
Short password
Confirm your password:
Log in automatically
Require my password to log in
Encrypt my home folder
Back
Continue
17:01

Create folder to make the Certificate and Key



Generate the Root CA Private Key : Generated a 2048-bit RSA private key for the Root Certificate Authority (CA). This key is encrypted with a passphrase for added security.
The private key (rootCA.key) will be used to sign both the server and client certificates later in the process.



Generate the Root CA Certificate: Generated a self-signed Root Certificate Authority (CA) certificate using the private key (rootCA.key) created earlier.

mintClient | mintServer

mohamed-elsayed@mohamed-elsayed-s: ~/cert_tls

```
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$ openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem
Enter pass phrase for rootCA.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]:EG
State or Province Name (full name) [Some-State]:Alexandria
Locality Name (eg, city) []:Alexandria
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AAST
Organizational Unit Name (eg, section) []:CyberSecurity
Common Name (e.g. server FQDN or YOUR name) []:mohamed221010750
Email Address []:mohamed9031515@gmail.com
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$
```

17:57

Generate the Server Private Key: Generated a 2048-bit RSA private key for the server (server.key).
Created a Certificate Signing Request (CSR) using the server's private key (server.key). The CSR contains the public key and the DN information, which will be used by the Root CA to sign the server certificate.

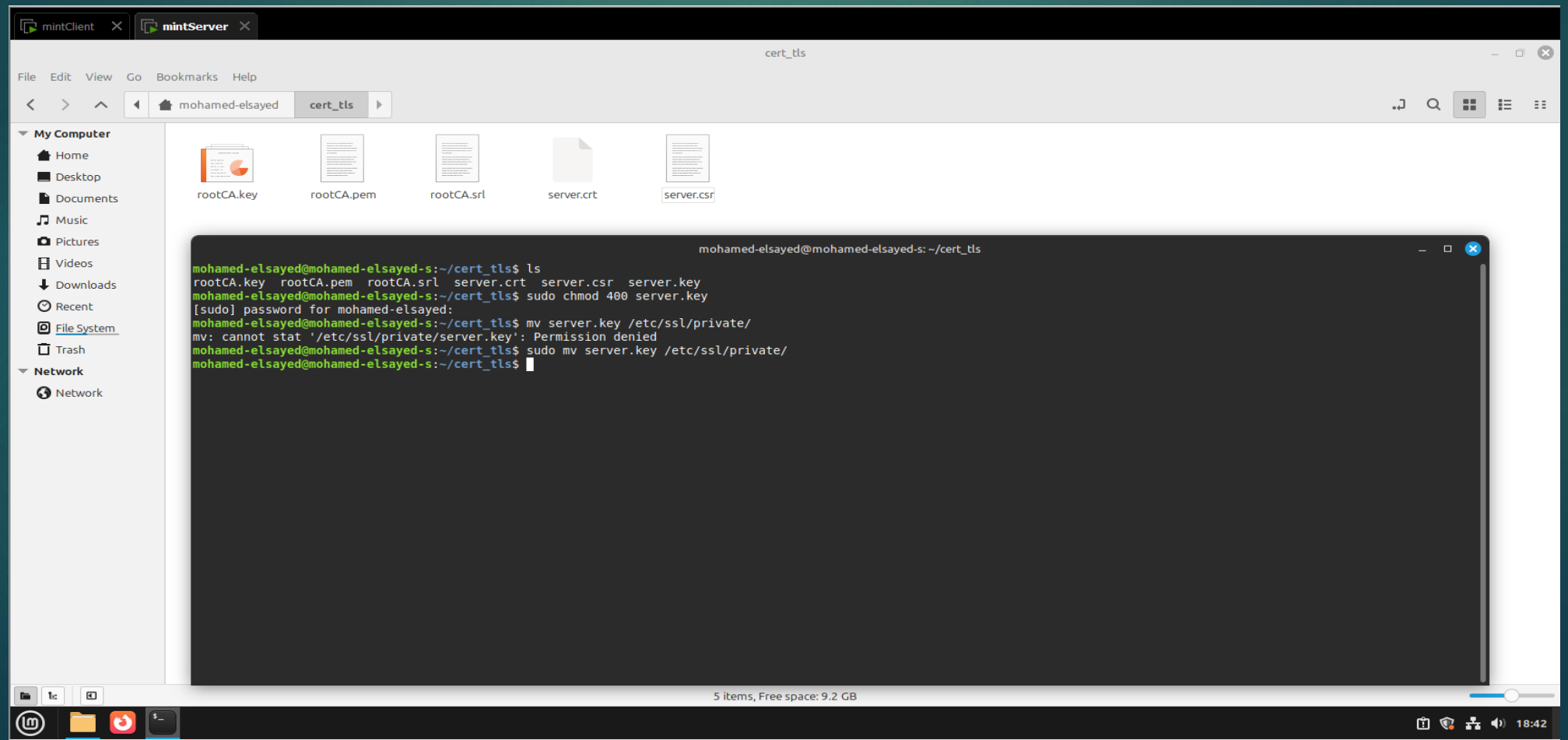
```
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$ openssl genrsa -out server.key 2048
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$ openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about 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]:EG
State or Province Name (full name) [Some-State]:Alexandria
Locality Name (eg, city) []:Alexandria
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AAST
Organizational Unit Name (eg, section) []:CyberSecurity
Common Name (e.g. server FQDN or YOUR name) []:mohamed221010750
Email Address []:mohamed9031515@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:mohamed221010750
An optional company name []:AAST
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$
```

Sign the Server Certificate with the Root CA:Signed the server's Certificate Signing Request (CSR) using the Root CA certificate (rootCA.pem) and its private key (rootCA.key).
Generated a valid server certificate (server.crt)

```
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$ openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out server.crt -days 365 -sha256
Certificate request self-signature ok
subject=C = EG, ST = Alexandria, L = Alexandria, O = AAST, OU = CyberSecurity, CN = mohamed221010750, emailAddress = mohamed9031515@gmail.com
Enter pass phrase for rootCA.key:
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$
```

Securely Manage the Server Private Key:Secured the server's private key (server.key) by changing its permissions to 400 using the chmod command.

Attempted to move the server's private key (server.key) to the /etc/ssl/private/ directory.
Encountered a permission denied error due to insufficient privileges.



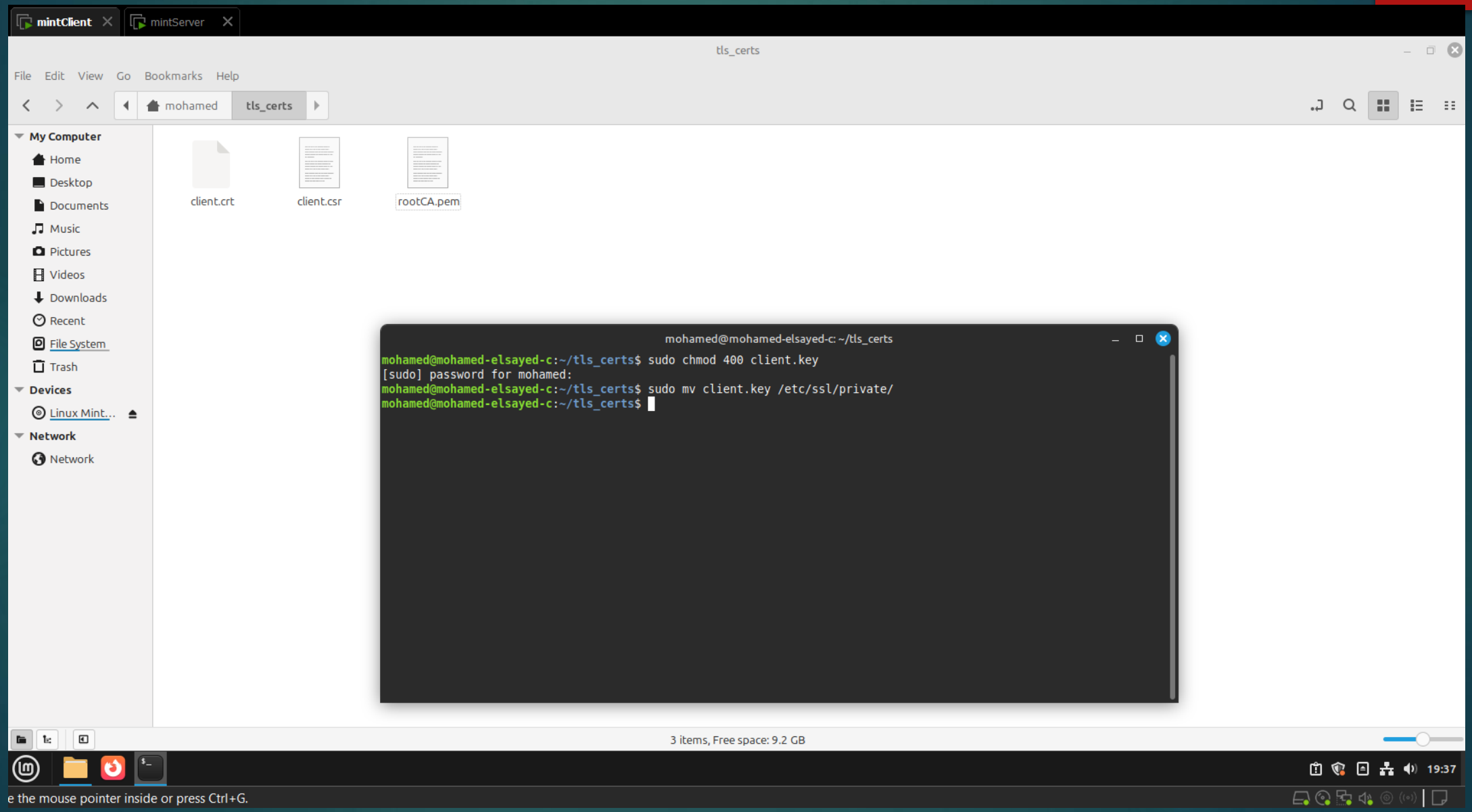
Generate the Client Private Key and Generate the Client Certificate Signing Request (CSR)

```
mohamed-elsayed@mohamed-elsayed-s: ~/cert_tls
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$ openssl genrsa -out client.key 2048
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$ openssl req -new -key client.key -out client.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]:EG
State or Province Name (full name) [Some-State]:Alexandria
Locality Name (eg, city) []:Alexandria
Organization Name (eg, company) [Internet Widgits Pty Ltd]:AAST
Organizational Unit Name (eg, section) []:CyberSecurity
Common Name (e.g. server FQDN or YOUR name) []:client
Email Address []:mohamed9031515@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:mohamed221010750
An optional company name []:AAST
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$
```

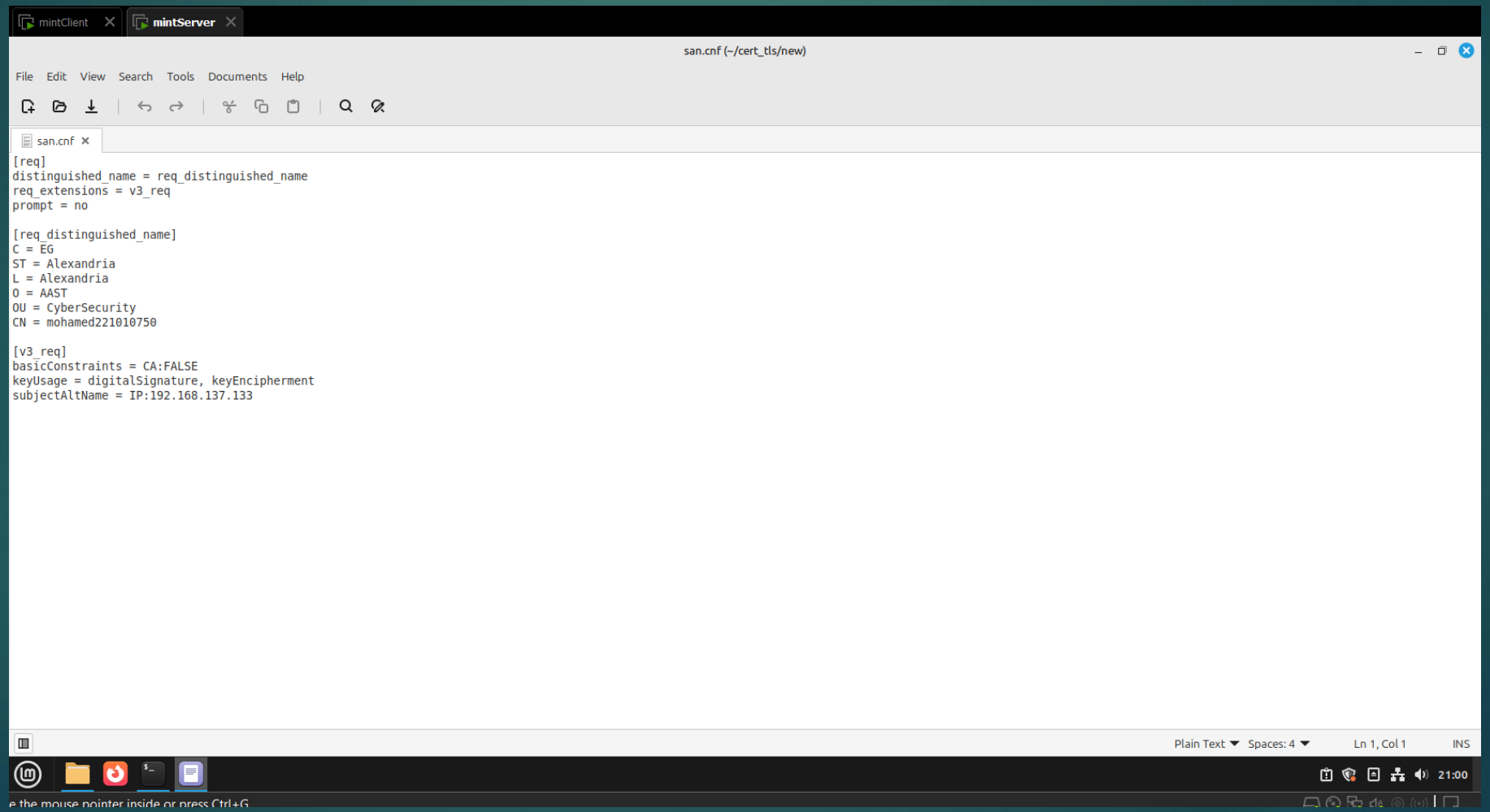
Sign the Client Certificate with the Root CA

```
mohamed-elsayed@mohamed-elsayed-s: ~/cert_tls
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$ openssl x509 -req -in client.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out client.crt -days 365 -sha256
Certificate request self-signature ok
subject=C = EG, ST = Alexandria, L = Alexandria, O = AAST, OU = CyberSecurity, CN = client, emailAddress = mohamed9031515@gmail.com
Enter pass phrase for rootCA.key:
mohamed-elsayed@mohamed-elsayed-s:~/cert_tls$
```

Securely Manage and Move the Client Private Key

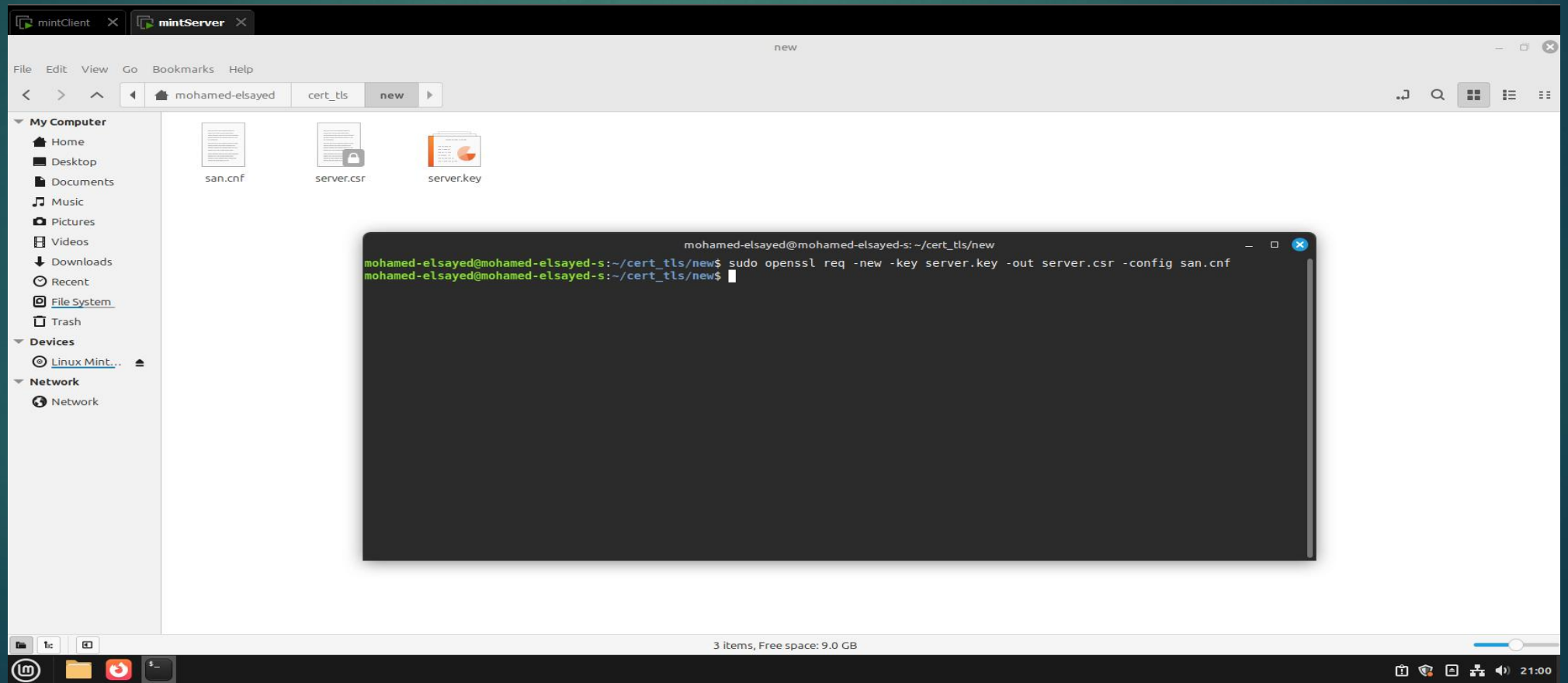


-Configure the san.cnf File for Certificate Generation: Created a configuration file (san.cnf) to define the Distinguished Name (DN) and other extensions for certificates.
-Defined the purpose of the certificate using keyUsage and added an alternative subject name (subjectAltName) for the IP address 192.168.137.133.



Generate a Server Certificate Signing Request (CSR) Using san.cnf :

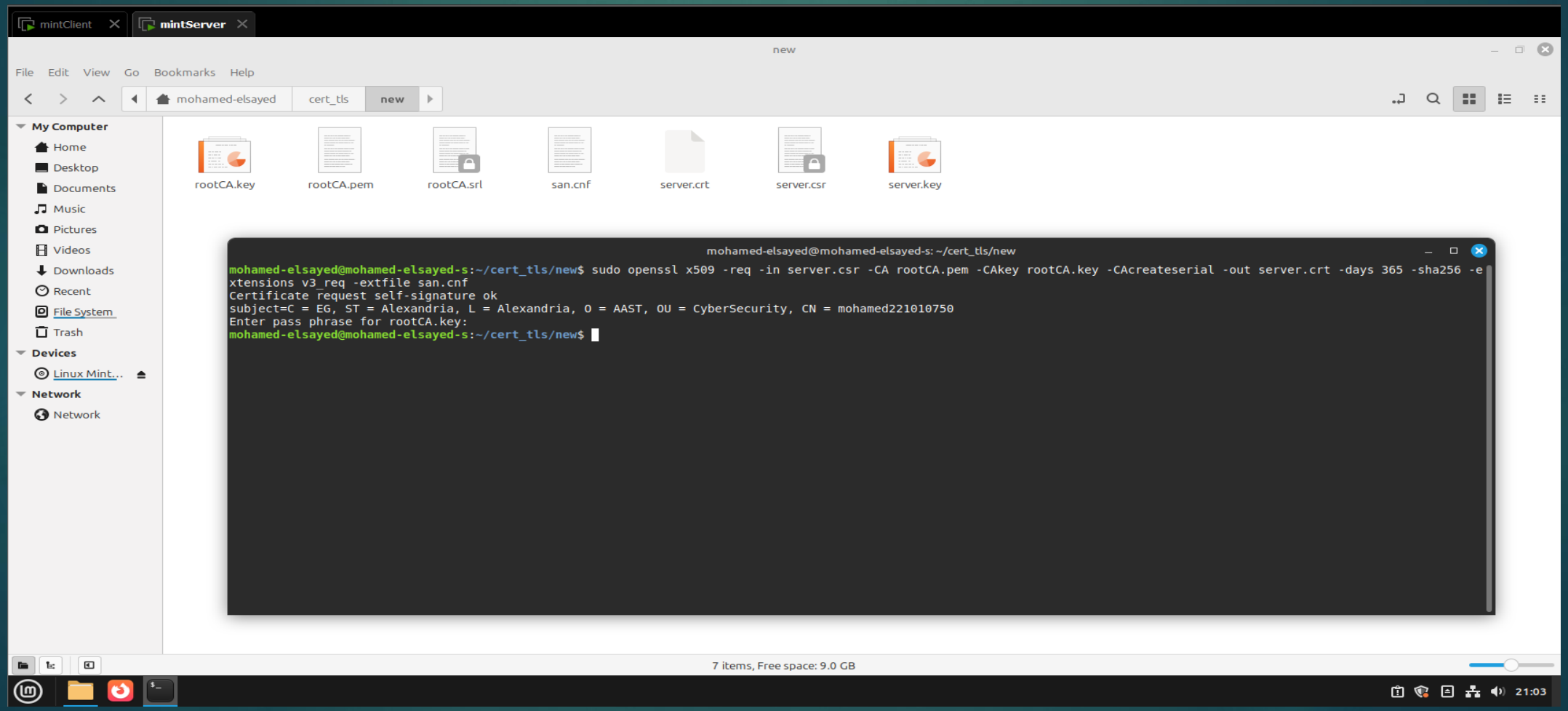
- Generated a Certificate Signing Request (CSR) for the server using the server.key private key and the san.cnf configuration file.
- Used the san.cnf file to ensure that the CSR includes the correct Distinguished Name (DN) and other extensions, such as the IP address SAN.
- Created the CSR file (server.csr), which will be signed by the Root CA to generate a valid server certificate.



Sign the Server Certificate with the Root CA:

-Signed the server's Certificate Signing Request (CSR) using the Root CA certificate (rootCA.pem) and its private key (rootCA.key).

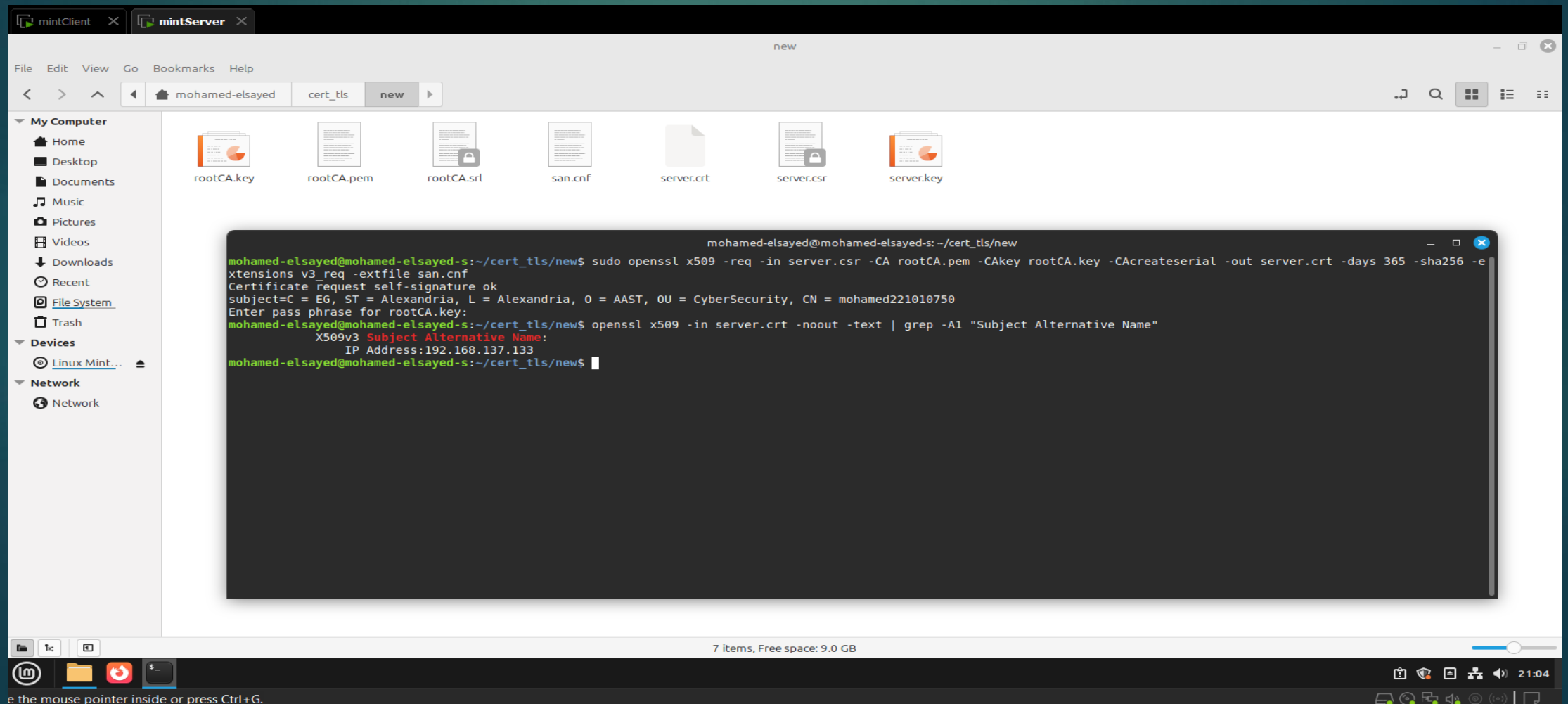
-Generated a valid server certificate (server.crt) with a validity period of 365 days.



Verify the Server Certificate's Subject Alternative Name (SAN):

-Verified the server certificate (server.crt) to ensure it includes the correct Subject Alternative Name (SAN).

-Used the grep command to filter the output and confirm that the IP address (192.168.137.133) is included in the SAN



Install python3-pip

```
mohamed-elsayed@mohamed-elsayed-s: ~/Desktop
mohamed-elsayed@mohamed-elsayed-s:~/Desktop$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  javascript-common libexpat1 libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython3-dev libpython3.12-dev
  libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 python3-dev python3-setuptools python3-wheel python3.12
  python3.12-dev python3.12-minimal zlib1g-dev
Suggested packages:
  apache2 | lighttpd | httpd python-setuptools-doc python3.12-venv python3.12-doc binfmt-support
The following NEW packages will be installed:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython3-dev libpython3.12-dev python3-dev
  python3-pip python3-setuptools python3-wheel python3.12-dev zlib1g-dev
The following packages will be upgraded:
  libexpat1 libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 python3.12 python3.12-minimal
6 upgraded, 13 newly installed, 0 to remove and 266 not upgraded.
Need to get 17.9 MB of archives.
After this operation, 43.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1 amd64 2.6.1-2ubuntu0.3 [88.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12t64 amd64 3.12.3-1ubuntu0.5 [2,339 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12 amd64 3.12.3-1ubuntu0.5 [651 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-stdlib amd64 3.12.3-1ubuntu0.5 [2,069 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 python3.12-minimal amd64 3.12.3-1ubuntu0.5 [2,342 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpython3.12-minimal amd64 3.12.3-1ubuntu0.5 [835 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 javascript-common all 11+nmu1 [5,936 B]
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1-dev amd64 2.6.1-2ubuntu0.3 [140 kB]
```

Build a Simple TLS Server Using Python

mintClient | mintServer

mohamed.py (~/Desktop)

File Edit View Search Tools Documents Help

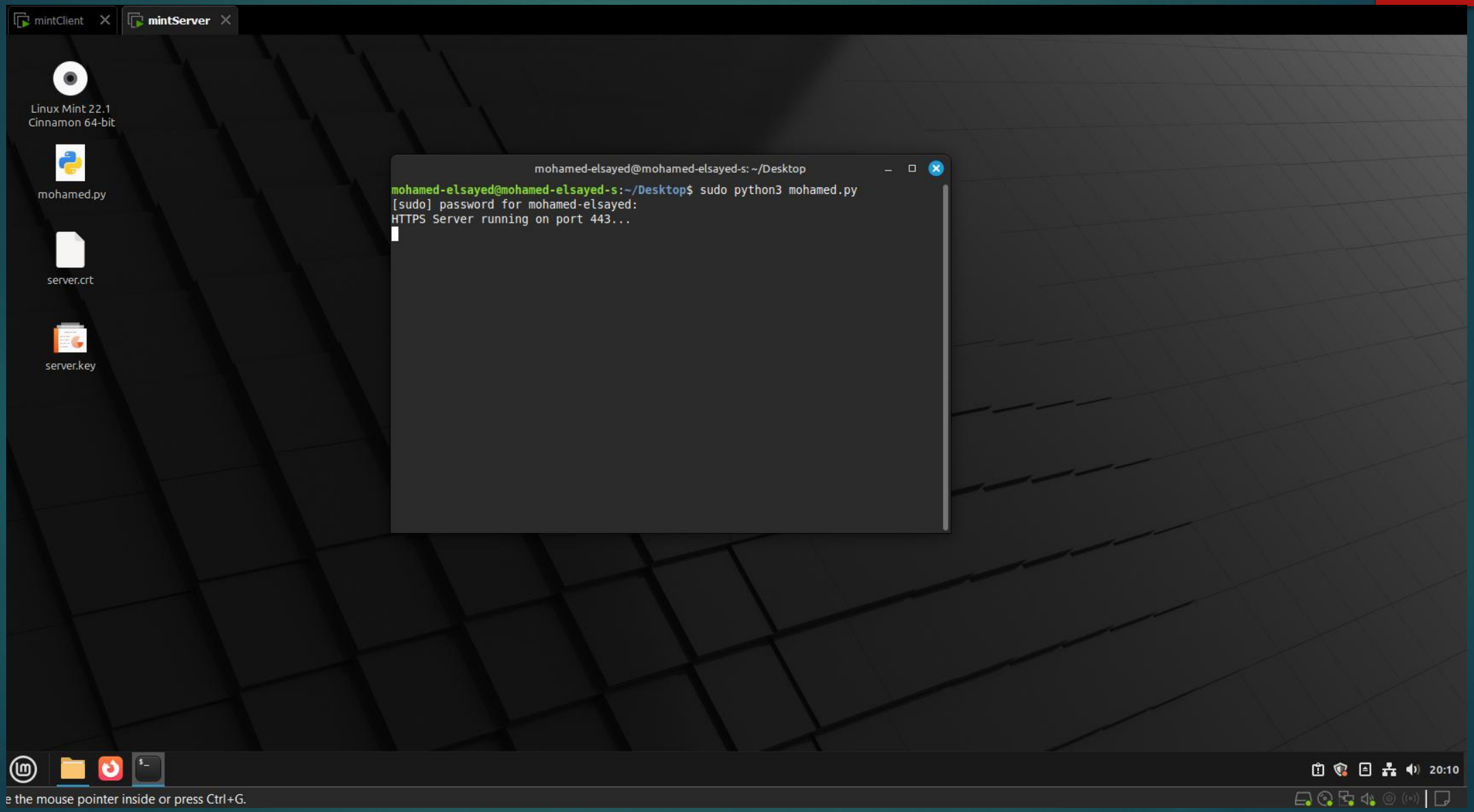
mohamed.py

```python
from http.server import HTTPServer, BaseHTTPRequestHandler
import ssl

httpd = HTTPServer(('0.0.0.0', 443), BaseHTTPRequestHandler)

context = ssl.SSLContext(ssl.PROTOCOL_TLS_SERVER)
context.load_cert_chain(certfile="server.crt", keyfile="server.key")

httpd.socket = context.wrap_socket(httpd.socket, server_side=True)

class SimpleHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(b"<h1>Mohamed's Secure Page</h1>")

httpd.RequestHandlerClass = SimpleHandler
print("HTTPS Server running on port 443...")
httpd.serve_forever()
```

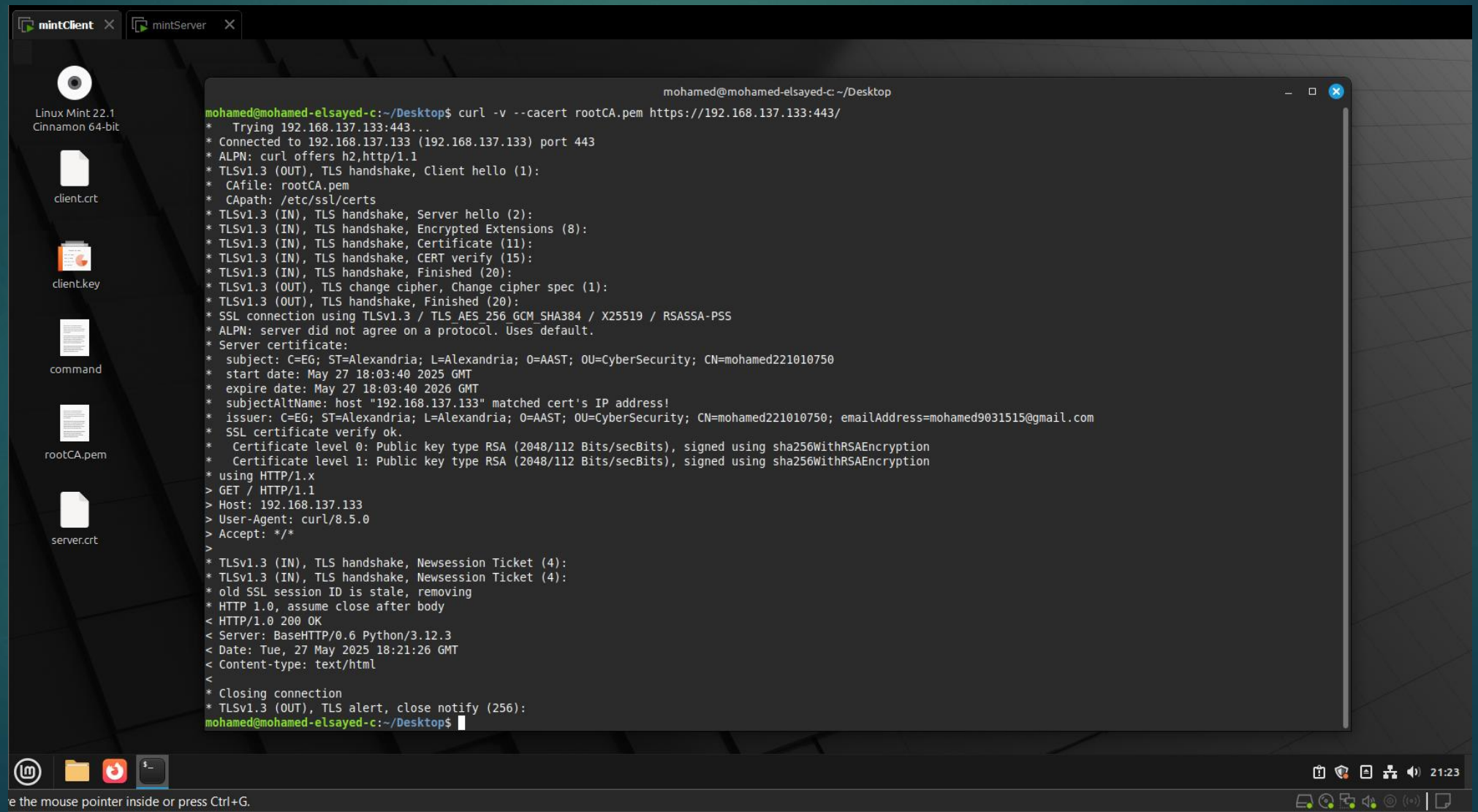
Python 2 Spaces: 4 Ln 13, Col 24 INS

20:10

Run the TLS Server Script



Test the TLS Connection Using curl



mintClient | mintServer

Settings — Mozilla Firefox

Settings

Firefox about:preferences#searchResults

Your browser is being managed by your organization.

cert

General
Home
Search
Privacy & Security
Sync
Firefox Labs
More from Mozilla

Extensions & Themes
Firefox Support

Search Results

Certificates

Query OCSP responder servers to confirm the current validity of certificates

Certificate Manager

Your Certificates | Authentication Decisions | People | Servers | Authorities

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
ACCV	
ACCVRAIZ1	Builtin Object Token
Actalis S.p.A./03358520967	
Actalis Authentication Root CA	Builtin Object Token
AffirmTrust	
AffirmTrust Premium ECC	Builtin Object Token
AffirmTrust Networking	Builtin Object Token

View... | Edit Trust... | Import... | Export... | Delete or Distrust...

OK

e the mouse pointer inside or press Ctrl+G.

21:23

mintClient | mintServer

Settings — Mozilla Firefox

Settings

Firefox about:preferences#privacy

Your browser is being managed by your organization.

Find in Settings

- General
- Home
- Search
- Privacy & Security
- Sync
- Firefox Labs
- More from Mozilla

Website Advertising Preferences

Allow websites to perform p
This helps sites understand how

Security

Deceptive Content and Da

Block dangerous and decepl
Block dangerous downlc
Warn you about unwant

Certificates

Query OCSP responder serv
certificates

HTTPS-Only Mode

Firefox creates secure and encrypted connections to sites you visit. Firefox will warn you if a connection isn't secure when HTTPS-Only is on.
Learn more

Enable HTTPS-Only Mode in all windows

Manage Exceptions...

- Extensions & Themes
- Firefox Support

Certificate Manager

Your Certificates | Authentication Decisions | People | Servers | Authorities

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
ACCV	
ACCVRAIZ1	Builtin Object Token
Actalis S.p.A./03358520967	
Actalis Authentication Root CA	Builtin Object Token
AffirmTrust	
AffirmTrust Premium ECC	Builtin Object Token
AffirmTrust Networking	Builtin Object Token

View... | Edit Trust... | Import... | Export... | Delete or Distrust...

OK

21:24

e the mouse pointer inside or press Ctrl+G.

mintClient | mintServer

Settings

Firefo

General
Home
Search
Privacy & Security
Sync
Firefox Labs
More from Mozilla

Extensions & Themes
Firefox Support

Select File containing CA certificate(s) to import

Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Linux Min...
Other Locations

mohamed | Desktop

Name	Size	Type	Modified
client.crt	1.4 kB	X.509 Certificate	19:21
rootCA.pem	1.5 kB	Text	17:57
server.crt	1.5 kB	X.509 Certificate	21:03

Certificate Files

Cancel | Open

21:26

e the mouse pointer inside or press Ctrl+G.

mintClient | mintServer

Settings — Mozilla Firefox

Settings

Firefox about:preferences#privacy

Your browser is being managed by your organization.

Find in Settings

General
Home
Search
Privacy & Security
Sync
Firefox Labs
More from Mozilla

Extensions & Themes
Firefox Support

Website Advertising Preferences

Allow websites to perform p
This helps sites understand

Security

Deceptive Content and

Block dangerous and de
Block dangerous dov
Warn you about unw

Certificates

Query OCSP responder serv
certificates

HTTPS-Only Mode

Firefox creates secure and encrypted connections to sites you visit. Firefox will warn you if a connection isn't secure when HTTPS-Only is on.
Learn more

Enable HTTPS-Only Mode in all windows

Manage Exceptions...

Certificate Manager

AffirmTrust Premium ECC | Builtin Object Token
AffirmTrust Networking | Builtin Object Token

View... | Edit Trust... | Import... | Export... | Delete or Distrust...

OK

Downloading Certificate

You have been asked to trust a new Certificate Authority (CA).

Do you want to trust "mohamed221010750" for the following purposes?

- [x] Trust this CA to identify websites.
- [x] Trust this CA to identify email users.

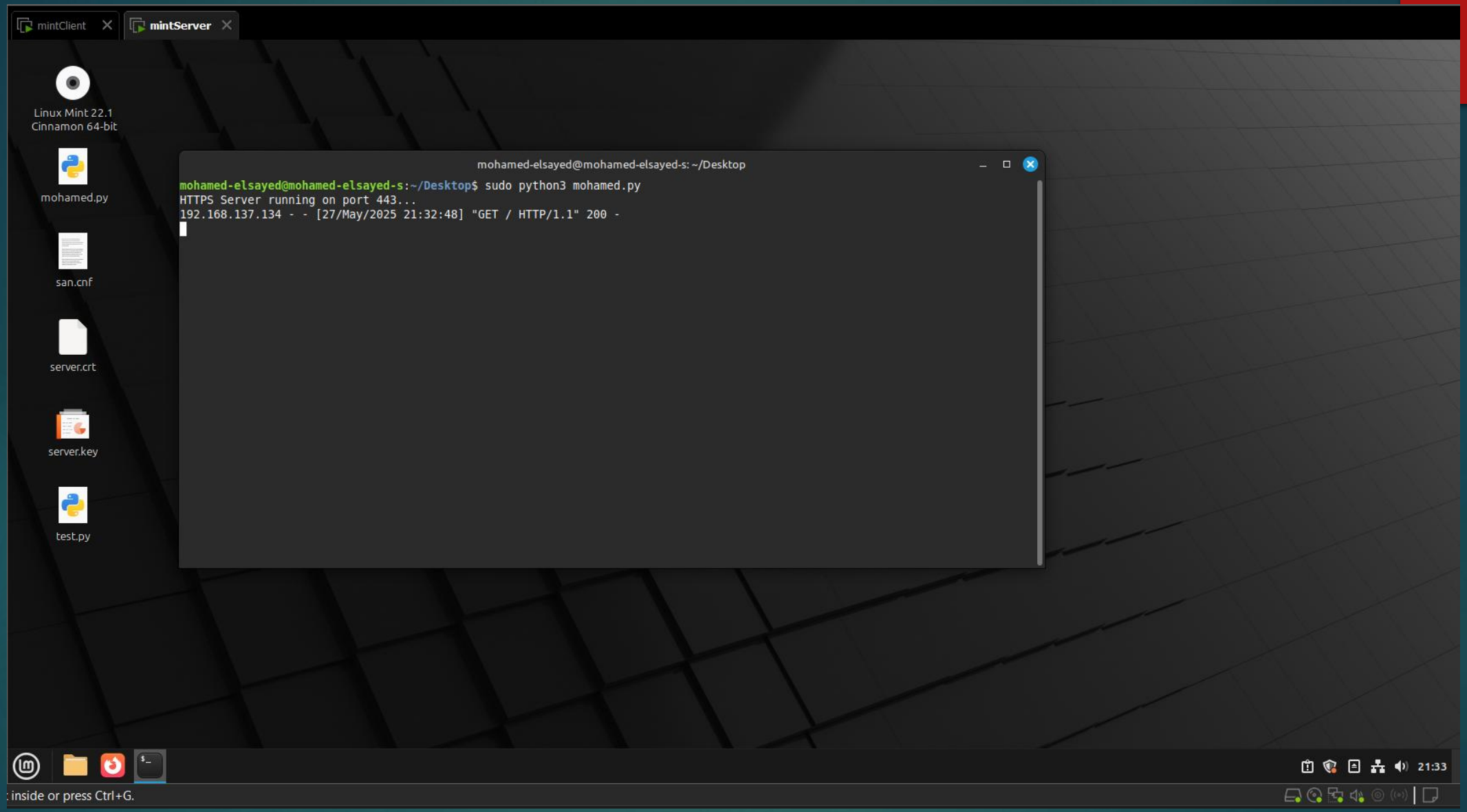
Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

View | Examine CA certificate

Cancel | OK

21:26

the mouse pointer inside or press Ctrl+G.


mintClient
mintServer
Linux Mint 22.1
Cinnamon 64-bit
mohamed.py
san.cnf
server.crt
server.key
test.py
mohamed-elsayed@mohamed-elsayed-s: ~/Desktop
mohamed-elsayed@mohamed-elsayed-s:~/Desktop$ sudo python3 mohamed.py
HTTPS Server running on port 443...
192.168.137.134 - - [27/May/2025 21:32:48] "GET / HTTP/1.1" 200 -
21:33
inside or press Ctrl+G.

mintClient mintServer

Mozilla Firefox

192.168.137.133/

https://192.168.137.133

Mohamed221010750
Omar221010339

21:33

e the mouse pointer inside or press Ctrl+G.

Install WireShark

mintClient | mintServer

Linux Mint 22.1
Cinnamon 64-bit

client.crt

client.key

command

rootCA.pem

server.crt

mohamed@mohamed-elsayed-c: ~/Desktop

```
mohamed@mohamed-elsayed-c:~/Desktop$ sudo apt install wireshark
[sudo] password for mohamed:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libb2-1 libbcg729-0 libcares2 libminizip1t64 libnghttp3-3 libqt6core5compat6 libqt6core6t64 libqt6dbus6t64 libqt6gui6t64 libqt6multimedia6 libqt6network6t64
  libqt6opengl6t64 libqt6printsupport6t64 libqt6qml6 libqt6qmlmodels6 libqt6quick6 libqt6svg6 libqt6waylandclient6 libqt6waylandcompositor6
  libqt6waylandeglclienthwintegration6 libqt6waylandeglcompositorhwintegration6 libqt6widgets6t64 libqt6wlshellintegration6 libsmi2t64 libts0t64 libwireshark-data
  libwireshark17t64 libwiretap14t64 libwsutil15t64 qt6-gtk-platformtheme qt6-qpa-plugins qt6-translations-l10n qt6-wayland wireshark-common
Suggested packages:
  qt6-qmltooling-plugins snmp-mibs-downloader geoipupdate geoip-database-extra libjs-leaflet libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
  libb2-1 libbcg729-0 libcares2 libminizip1t64 libnghttp3-3 libqt6core5compat6 libqt6core6t64 libqt6dbus6t64 libqt6gui6t64 libqt6multimedia6 libqt6network6t64
  libqt6opengl6t64 libqt6printsupport6t64 libqt6qml6 libqt6qmlmodels6 libqt6quick6 libqt6svg6 libqt6waylandclient6 libqt6waylandcompositor6
  libqt6waylandeglclienthwintegration6 libqt6waylandeglcompositorhwintegration6 libqt6widgets6t64 libqt6wlshellintegration6 libsmi2t64 libts0t64 libwireshark-data
  libwireshark17t64 libwiretap14t64 libwsutil15t64 qt6-gtk-platformtheme qt6-qpa-plugins qt6-translations-l10n qt6-wayland wireshark wireshark-common
0 upgraded, 35 newly installed, 0 to remove and 272 not upgraded.
Need to get 46.7 MB/46.8 MB of archives.
After this operation, 211 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 libb2-1 amd64 0.98.1-1.1build1 [45.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 libbcg729-0 amd64 1.1.1-2build1 [35.3 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 libminizip1t64 amd64 1:1.3.dfsg-3.1ubuntu2.1 [22.2 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 libnghttp3-3 amd64 0.8.0-2 [59.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 libqt6core6t64 amd64 6.4.2+dfsg-21.1build5 [1,789 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 libqt6core5compat6 amd64 6.4.2-4build3 [147 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 libqt6dbus6t64 amd64 6.4.2+dfsg-21.1build5 [267 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 libts0t64 amd64 1.22-1.1build1 [63.8 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 libqt6gui6t64 amd64 6.4.2+dfsg-21.1build5 [3,119 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/universe amd64 libqt6network6t64 amd64 6.4.2+dfsg-21.1build5 [740 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/universe amd64 libqt6multimedia6 amd64 6.4.2-11build3 [527 kB]
```

21:38

e the mouse pointer inside or press Ctrl+G.

mintClient | mintServer

*ens33

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 443 && tls

No.	Time	Source	Destination	Protocol	Length	Info
84	177.236978670	192.168.137.134	192.168.137.133	TLSv1.3	583	Client Hello
86	177.247948692	192.168.137.133	192.168.137.134	TLSv1.3	1671	Server Hello, Change Cipher Spec, Appli
88	177.250686991	192.168.137.134	192.168.137.133	TLSv1.3	146	Change Cipher Spec, Application Data
89	177.250847701	192.168.137.134	192.168.137.133	TLSv1.3	166	Application Data
90	177.251676874	192.168.137.133	192.168.137.134	TLSv1.3	321	Application Data
91	177.252339385	192.168.137.133	192.168.137.134	TLSv1.3	524	Application Data, Application Data, App
93	177.252575988	192.168.137.134	192.168.137.133	TLSv1.3	90	Application Data

```
▸ Frame 84: 583 bytes on wire (4664 bits), 583 bytes captu
▸ Ethernet II, Src: VMware_20:f2:fa (00:0c:29:20:f2:fa), D
▸ Internet Protocol Version 4, Src: 192.168.137.134, Dst:
▸ Transmission Control Protocol, Src Port: 53856, Dst Port
▸ Transport Layer Security
```

```
0000  00 0c 29 ee 96 5e 00 0c  29 20 f2 fa 08 00 45 00
0010  02 39 05 01 40 00 40 06  9f 61 c0 a8 89 86 c0 a8
0020  89 85 d2 60 01 bb bf d7  af 20 1d 21 1f 03 80 18
0030  01 f6 96 88 00 00 01 01  08 0a a6 96 e6 17 72 17
0040  f5 5f 16 03 01 02 00 01  00 01 fc 03 03 81 6d 1f
0050  24 93 f1 1c f8 bf 4d 17  15 e0 04 3c 4f 77 da 92
0060  5e 1f be 7d 26 9c e8 12  17 05 74 40 70 20 da a9
0070  2f 7d 2b f1 ba 9d c0 f1  3a e5 36 0d b0 6b 07 db
0080  1e a9 e2 f8 82 fe ac e5  3d 61 cf 2a 5a b8 00 3e
0090  13 02 13 03 13 01 c0 2c  c0 30 00 9f cc a9 cc a8
00a0  cc aa c0 2b c0 2f 00 9e  c0 24 c0 28 00 6b c0 23
00b0  c0 27 00 67 c0 0a c0 14  00 39 c0 09 c0 13 00 33
00c0  00 9d 00 9c 00 3d 00 3c  00 35 00 2f 00 ff 01 00
00d0  01 75 00 0b 00 04 03 00  01 02 00 0a 00 16 00 14
00e0  00 1d 00 17 00 1e 00 19  00 18 01 00 01 01 01 02
00f0  01 03 01 04 00 10 00 0e  00 0c 02 68 32 08 68 74
0100  74 70 2f 31 2e 31 00 16  00 00 00 17 00 00 00 31
0110  00 00 00 0d 00 2a 00 28  04 03 05 03 06 03 08 07
0120  08 08 08 09 08 0a 08 0b  08 04 08 05 08 06 04 01
0130  05 01 06 01 03 03 03 01  03 02 04 02 05 02 06 02
0140  00 2b 00 05 04 03 04 03  03 00 2d 00 02 01 01 00
0150  33 00 26 00 24 00 1d 00  20 02 0c ce 1a 67 30 cd
```

wireshark_ens33RZNK72.pcapng Packets: 98 · Displayed: 7 (7.1%) Profile: Default

mohamed@mohamed-elsayed-c: ~/Desktop

```
mohamed@mohamed-elsayed-c:~/Desktop$ curl -v --cacert rootCA.pem https://192.168.137.133:443/
*   Trying 192.168.137.133:443...
* Connected to 192.168.137.133 (192.168.137.133) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: rootCA.pem
*  CApath: /etc/ssl/certs
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / X25519 / RSASSA-PSS
* ALPN: server did not agree on a protocol. Uses default.
* Server certificate:
*  subject: C=EG; ST=Alexandria; L=Alexandria; O=AAST; OU=CyberSecurity; CN=mohamed221010750
*  start date: May 27 18:03:40 2025 GMT
*  expire date: May 27 18:03:40 2026 GMT
*  subjectAltName: host "192.168.137.133" matched cert's IP address!
*  issuer: C=EG; ST=Alexandria; L=Alexandria; O=AAST; OU=CyberSecurity; CN=mohamed221010750;
emailAddress=mohamed9031515@gmail.com
*  SSL certificate verify ok.
*   Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256With
RSAEncryption
*   Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256With
RSAEncryption
* using HTTP/1.x
> GET / HTTP/1.1
> Host: 192.168.137.133
> User-Agent: curl/8.5.0
> Accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* old SSL session ID is stale, removing
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: BaseHTTP/0.6 Python/3.12.3
< Date: Wed, 28 May 2025 19:42:55 GMT
< Content-type: text/html
<
* Closing connection
* TLSv1.3 (OUT), TLS alert, close notify (256):
<h1>Mohamed221010750<br>Omar221010339</h1>mohamed@mohamed-elsayed-c:~/Desktop$
```

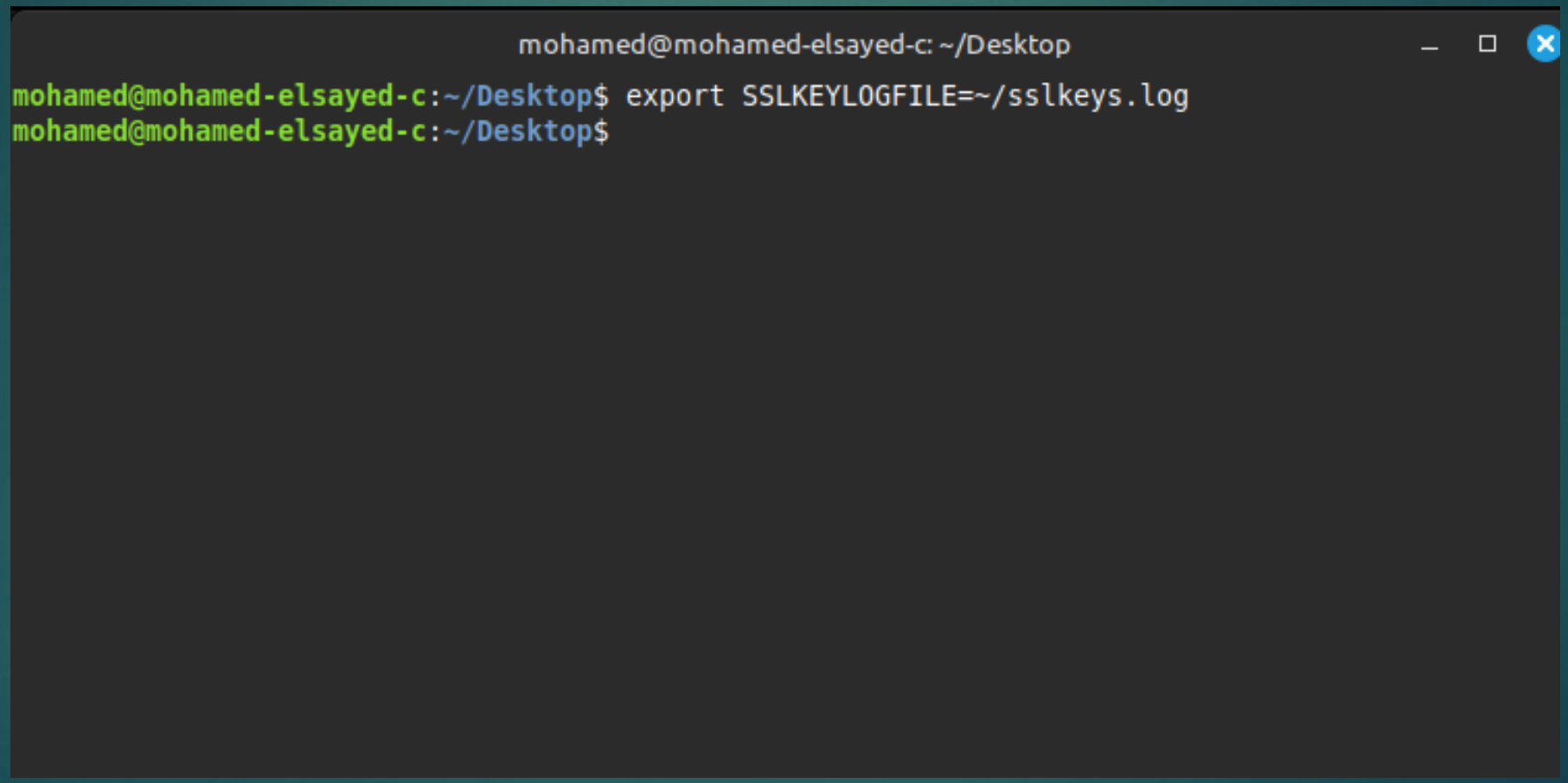
22:43

e the mouse pointer inside or press Ctrl+G.

Set Up SSL Key Logging for Wireshark Decryption:

-Set the SSLKEYLOGFILE environment variable to /sslkeys.log.

-Prepared the system to log SSL/TLS session keys, which will be used later to decrypt captured TLS traffic in Wireshark.



mintClient | mintServer

Capturing from ens33

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 443 && tls

No.	Time	Source	Destination	Protocol	Length	Info
4	0.002161026	192.168.137.134	192.168.137.133			
6	0.003726304	192.168.137.133	192.168.137.134			
8	0.004941995	192.168.137.134	192.168.137.133			
9	0.005188669	192.168.137.134	192.168.137.133			
10	0.005502884	192.168.137.133	192.168.137.134			
11	0.006083186	192.168.137.133	192.168.137.134			
13	0.006269365	192.168.137.134	192.168.137.133			

```
Frame 4: 583 bytes on wire (4664 bits), 583 bytes captur
Ethernet II, Src: VMware_20:f2:fa (00:0c:29:20:f2:fa), D
Internet Protocol Version 4, Src: 192.168.137.134, Dst:
Transmission Control Protocol, Src Port: 58504, Dst Port
Transport Layer Security
```

```
0130  05 01 06 01 03 03 03 01  03 02 04 02 05 02 06 02
0140  00 2b 00 05 04 03 04 03  03 00 2d 00 02 01 01 00
0150  33 00 26 00 24 00 1d 00  20 ff 08 93 b1 0e 11 92
0160  88 60 c6 a8 92 cd 7a 06  a6 8c 8b e9 bd 95 ae b8
0170  42 27 d9 e2 60 65 34 eb  38 00 15 00 ca 00 00 00
```

Transmission Control Protocol (tcp), ٣٢ bytes | Packets: 25 · Displayed: 7 (28.0%) | Profile: Default

Wireshark · Preferences

TIPC
TiVoConnect
TLS
TNS
Token-Ring
TPCP
TPKT
TPLINK-SMA..
TPM2.0
TPNCP
TRANSUM
TSDNS
TSP
TTE
TURNCHANN
TUXEDO
TZSP
UA3G
UASIP
UAUDP
UBDP
UBERTOOTH
UCI
UCP
UDP
UDP-Lite
UDPCP
UDPENCAP
UDS
UDT
UFTP

Transport Layer Security

RSA keys list Edit...

TLS debug file Browse...

☑ Reassemble TLS records spanning multiple TCP segments

☑ Reassemble TLS Application Data spanning multiple TLS records

☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared Key

(Pre)-Master-Secret log filename

/home/mohamed/ssl_keys.log Browse...

Help | Cancel | OK

mohamed@mohamed-elsayed-c: ~/Desktop

```
mohamed@mohamed-elsayed-c:~/Desktop$ curl -v --cacert rootCA.pem https://192.168.137.133:443
*   Trying 192.168.137.133:443...
* Connected to 192.168.137.133 (192.168.137.133) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: rootCA.pem
*  CApath: /etc/ssl/certs
erver hello (2):
Encrypted Extensions (8):
ertificate (11):
ERT verify (15):
inished (20):
er, Change cipher spec (1):
Finished (20):
/ TLS_AES_256_GCM_SHA384 / X25519 / RSASSA-PSS
a protocol. Uses default.

L=Alexandria; O=AAST; OU=CyberSecurity; CN=mohamed221010750
2025 GMT
2026 GMT
.137.133" matched cert's IP address!
L=Alexandria; O=AAST; OU=CyberSecurity; CN=mohamed221010750;
l.com

key type RSA (2048/112 Bits/secBits), signed using sha256With

key type RSA (2048/112 Bits/secBits), signed using sha256With

ewsession Ticket (4):
ewsession Ticket (4):
emoving
ody

12.3
1 GMT
< Content-type: text/html
<
* Closing connection
* TLSv1.3 (OUT), TLS alert, close notify (256):
<h1>Mohamed221010750<br>Omar221010339</h1>mohamed@mohamed-elsayed-c:~/Desktop$
```

22:55

e the mouse pointer inside or press Ctrl+G.

mintClient | mintServer

*ens33

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 443 && tls

No.	Time	Source	Destination	Protocol	Length	Info
4	0.002161026	192.168.137.134	192.168.137.133	TLSv1.3	583	Client Hello
6	0.003726304	192.168.137.133	192.168.137.134	TLSv1.3	1671	Server Hello, Change Cipher Spec, Enc
8	0.004941995	192.168.137.134	192.168.137.133	TLSv1.3	146	Change Cipher Spec, Finished
9	0.005188669	192.168.137.134	192.168.137.133	HTTP	166	GET / HTTP/1.1
10	0.005502884	192.168.137.133	192.168.137.134	TLSv1.3	321	New Session Ticket
11	0.006083186	192.168.137.133	192.168.137.134	HTTP	524	HTTP/1.0 200 OK (text/html)
13	0.006269365	192.168.137.134	192.168.137.133	TLSv1.3	90	Alert (Level: Warning, Description: C

wireshark_ens33YKGL72.pcapng Packets: 29 · Displayed: 7 (24.1%) Profile: Default

mohamed@mohamed-elsayed-c: ~/Desktop

```
mohamed@mohamed-elsayed-c:~/Desktop$ curl -v --cacert rootCA.pem https://192.168.137.133:443
*   Trying 192.168.137.133:443...
* Connected to 192.168.137.133 (192.168.137.133) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: rootCA.pem
*  CApath: /etc/ssl/certs
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / X25519 / RSASSA-PSS
* ALPN: server did not agree on a protocol. Uses default.
* Server certificate:
*  subject: C=EG; ST=Alexandria; L=Alexandria; O=AAST; OU=CyberSecurity; CN=mohamed221010750
*  start date: May 27 18:03:40 2025 GMT
*  expire date: May 27 18:03:40 2026 GMT
*  subjectAltName: host "192.168.137.133" matched cert's IP address!
*  issuer: C=EG; ST=Alexandria; L=Alexandria; O=AAST; OU=CyberSecurity; CN=mohamed221010750;
emailAddress=mohamed9031515@gmail.com
*  SSL certificate verify ok.
*   Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256With
RSAEncryption
*   Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256With
RSAEncryption
* using HTTP/1.x
> GET / HTTP/1.1
> Host: 192.168.137.133
> User-Agent: curl/8.5.0
> Accept: */*
>
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* old SSL session ID is stale, removing
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: BaseHTTP/0.6 Python/3.12.3
< Date: Wed, 28 May 2025 19:54:41 GMT
< Content-type: text/html
<
* Closing connection
* TLSv1.3 (OUT), TLS alert, close notify (256):
<h1>Mohamed221010750<br>Omar221010339</h1>mohamed@mohamed-elsayed-c:~/Desktop$
```

22:55

ve the mouse pointer inside or press Ctrl+G.