INTRODUCTION TO SSL

- Secure socket layer- encrypts the data stream between the web server and the web client (browser)
- Uses public key cryptography (PKI)
    - Private, public key
- Certificate signed by certificate authority – prevents impersonation attacks
    - SSL X.509
    - Self-signed certificate

OBJECTIVE: creating a self-signed certificate and encrypting the data stream between a web server and a web client (website)

TOOLS

- OpenSSL- general purpose cryptography library

TASK 1 : BASIC ENCRYPTION AND DECRYPTION USING OPEN SSL

- Using various encryption algorithms and modes on the text file and image file given
- Save each one separately and observe

QUESTIONS

1. Perform 2 types of encryptions on text_file.txt using different cipher types, compare the outputs. Mention the commands used for encryption and decryption, provide screenshots of the encrypted text. (cat (filename))

CyberSec-Server-2022 (2) - VMware Workstation 16 Player (Non-commercial use only)

Player

Terminal File Edit View Search Terminal Help 10:32 PM

sophiapineda - cybersec-server@ubuntu: ~/Desktop

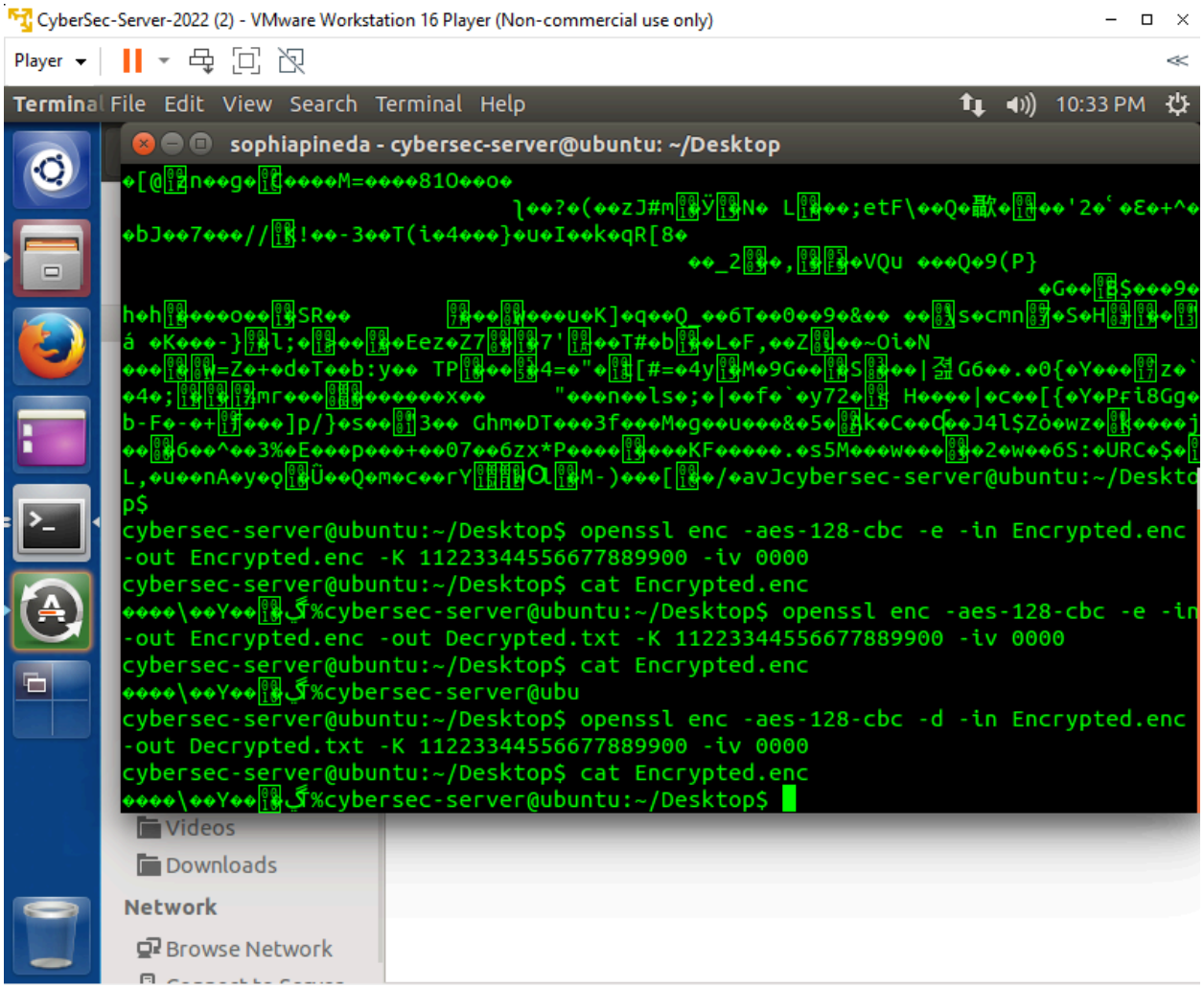
```
cybersec-server@ubuntu:~$ pwd
/home/cybersec-server
cybersec-server@ubuntu:~$ cd Desktop
cybersec-server@ubuntu:~/Desktop$ ls
cybersec-server@ubuntu:~/Desktop$ ls
cybersec-server@ubuntu:~/Desktop$ ls
image.bmp  Text_file.txt
cybersec-server@ubuntu:~/Desktop$ openssl enc -aes-128-cbc -e -in Text_file.txt
-out Encrypted.enc -K 11223344556677889900 -iv 0000
cybersec-server@ubuntu:~/Desktop$ cat Encrypted.enc
```

Videos

Downloads

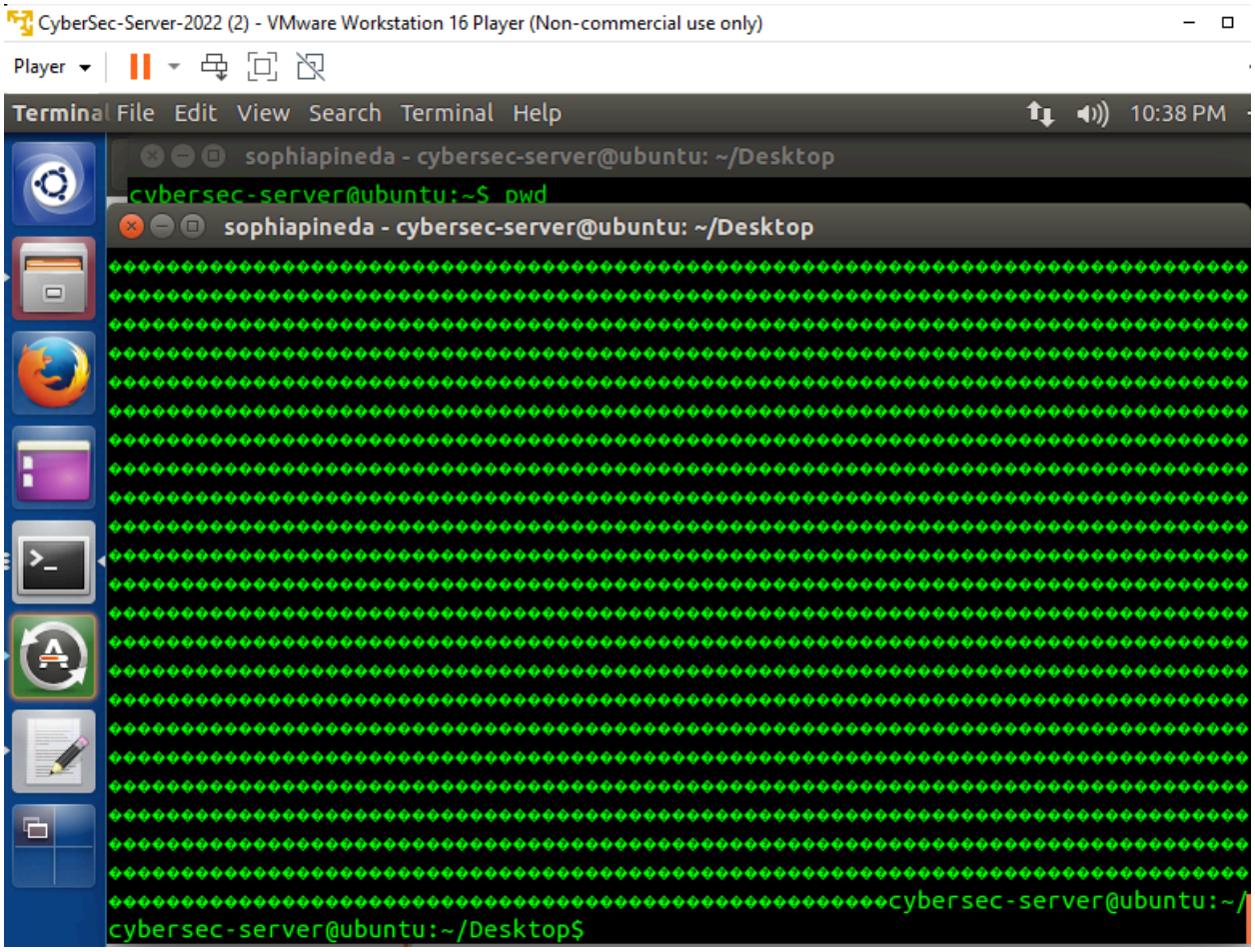
Network

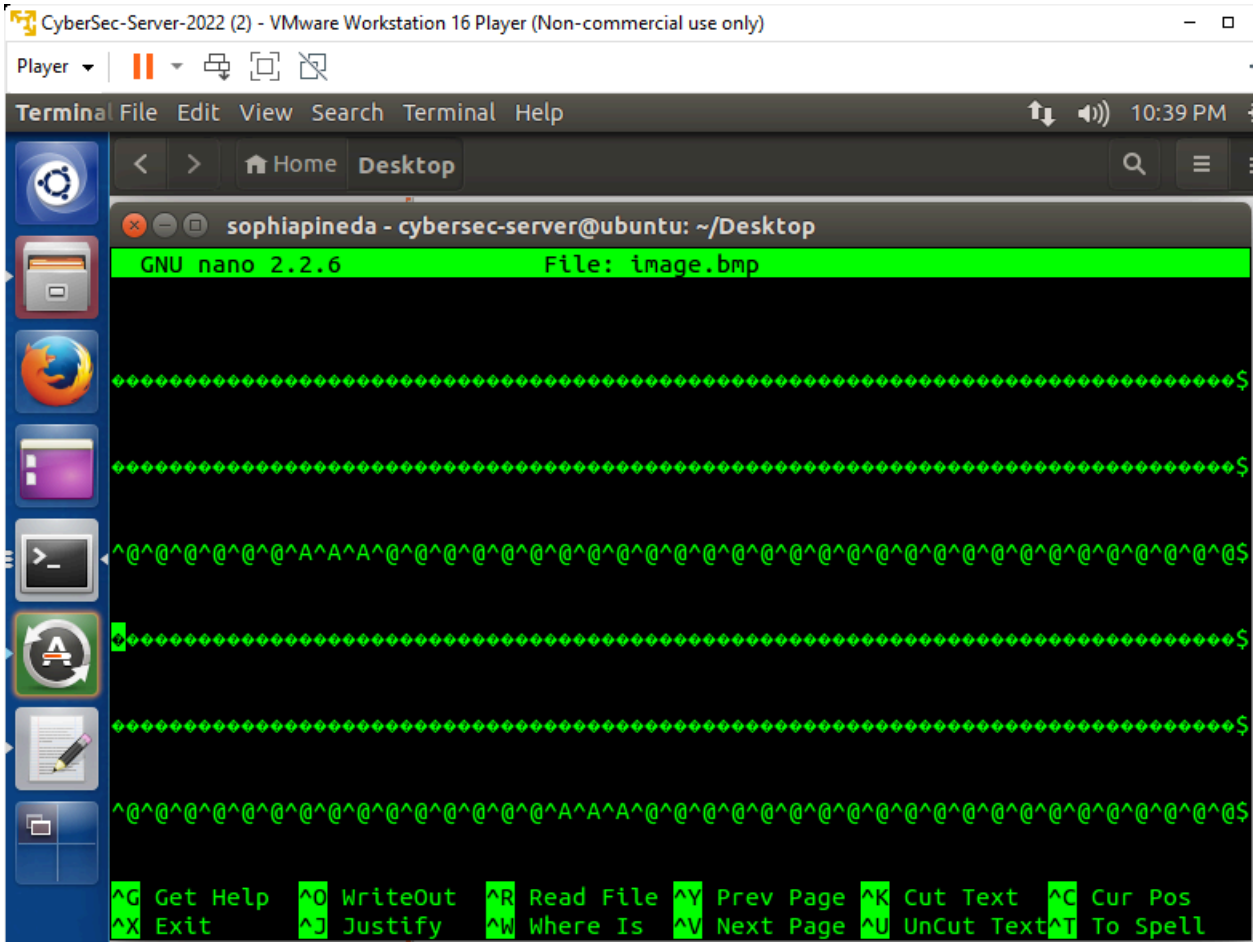
Browse Network

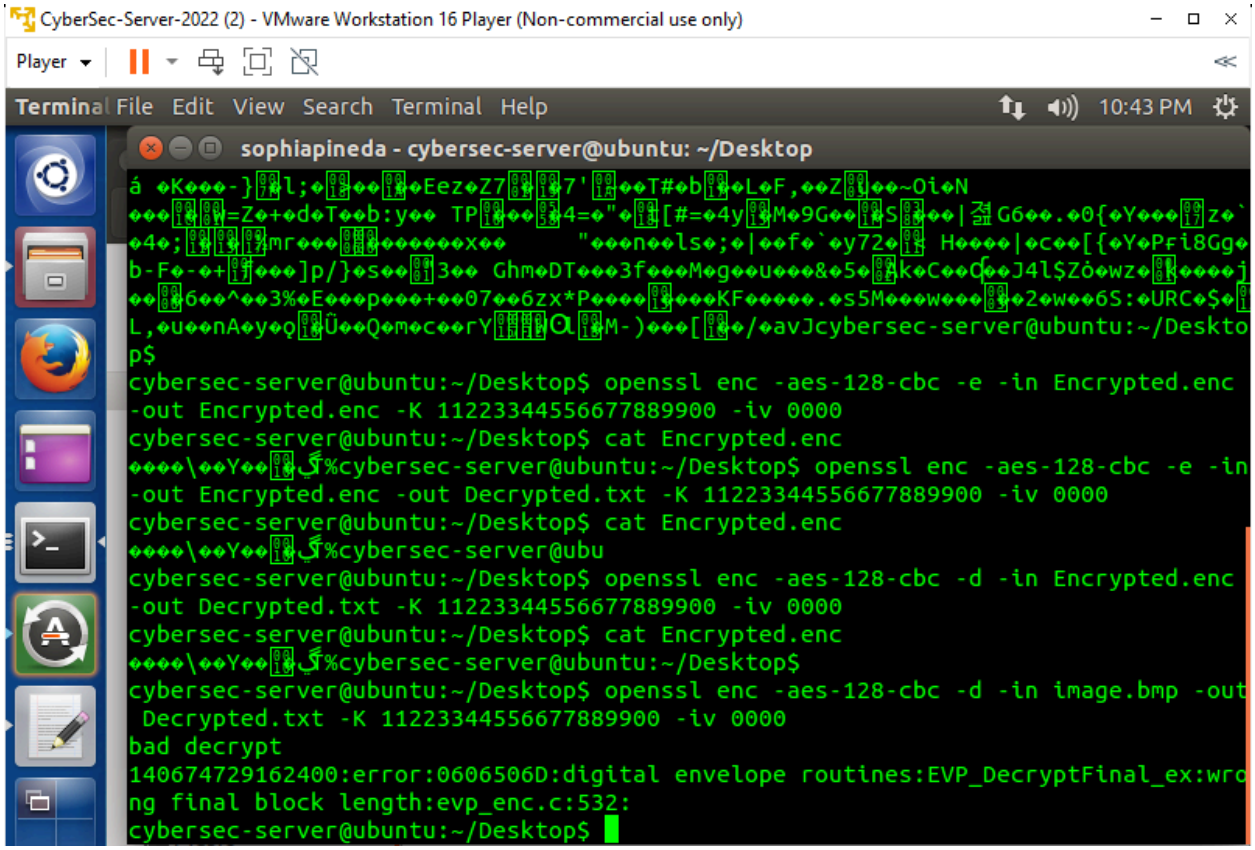


- Repeating patterns
- Scrambled letters

2. Perform 2 types of encryptions on image.bmp using different cipher types, compare the outputs. Mention the commands used for encryption and decryption, provide screenshots of the encrypted text. (sudo nano (filename))


CyberSec-Server-2022 (2) - VMware Workstation 16 Player (Non-commercial use only)
Player
Terminal File Edit View Search Terminal Help
10:38 PM
sophiapineda - cybersec-server@ubuntu: ~/Desktop
cybersec-server@ubuntu:~$ pwd
sophiapineda - cybersec-server@ubuntu: ~/Desktop
cybersec-server@ubuntu:~/
cybersec-server@ubuntu:~/Desktop$


CyberSec-Server-2022 (2) - VMware Workstation 16 Player (Non-commercial use only)
Player
Terminal File Edit View Search Terminal Help
10:39 PM
Home Desktop
sophiapineda - cybersec-server@ubuntu: ~/Desktop
GNU nano 2.2.6 File: image.bmp
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell



- Unable to decrypt
- Question marks
- Hashes?

TASK 2: BECOME A CERTIFICATE AUTHORITY

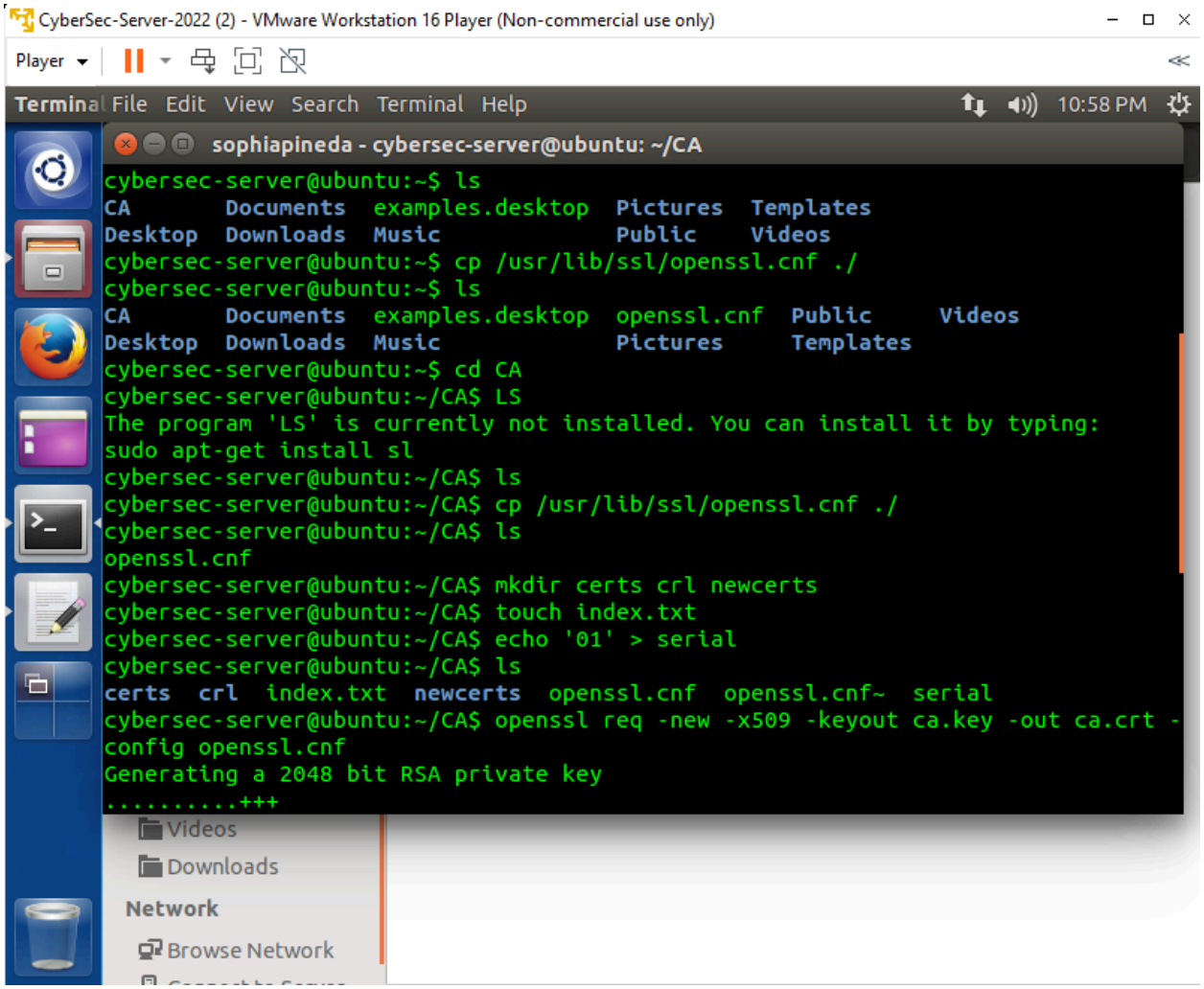
- Becoming a root CA to issue others certificates (servers)- self signed

STEPS:

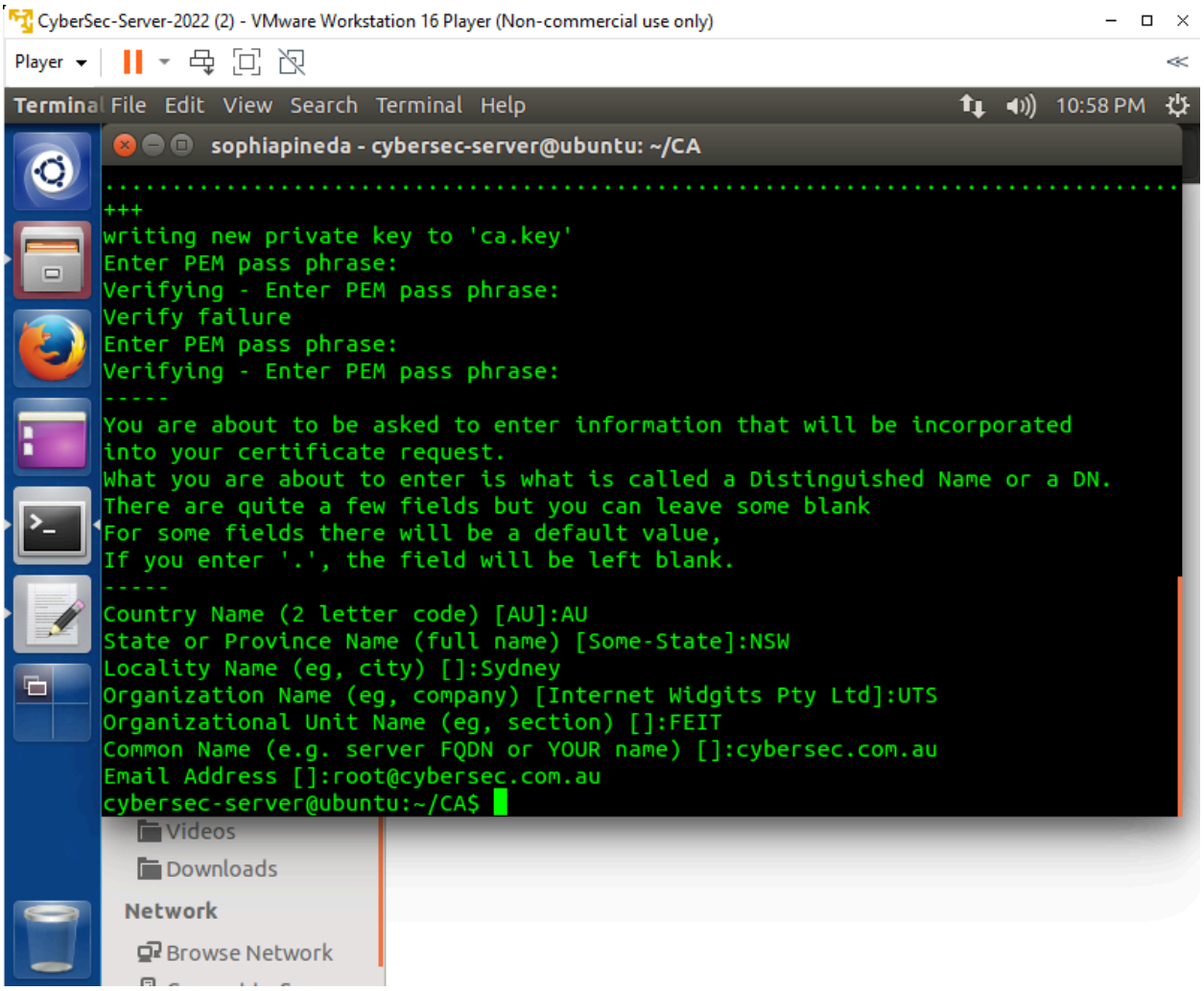
- Create directory
- Copy file and move it into folder
- Modify file
- Type in command to make the certificate
- Type in command to decrypt content

SCREENSHOTS

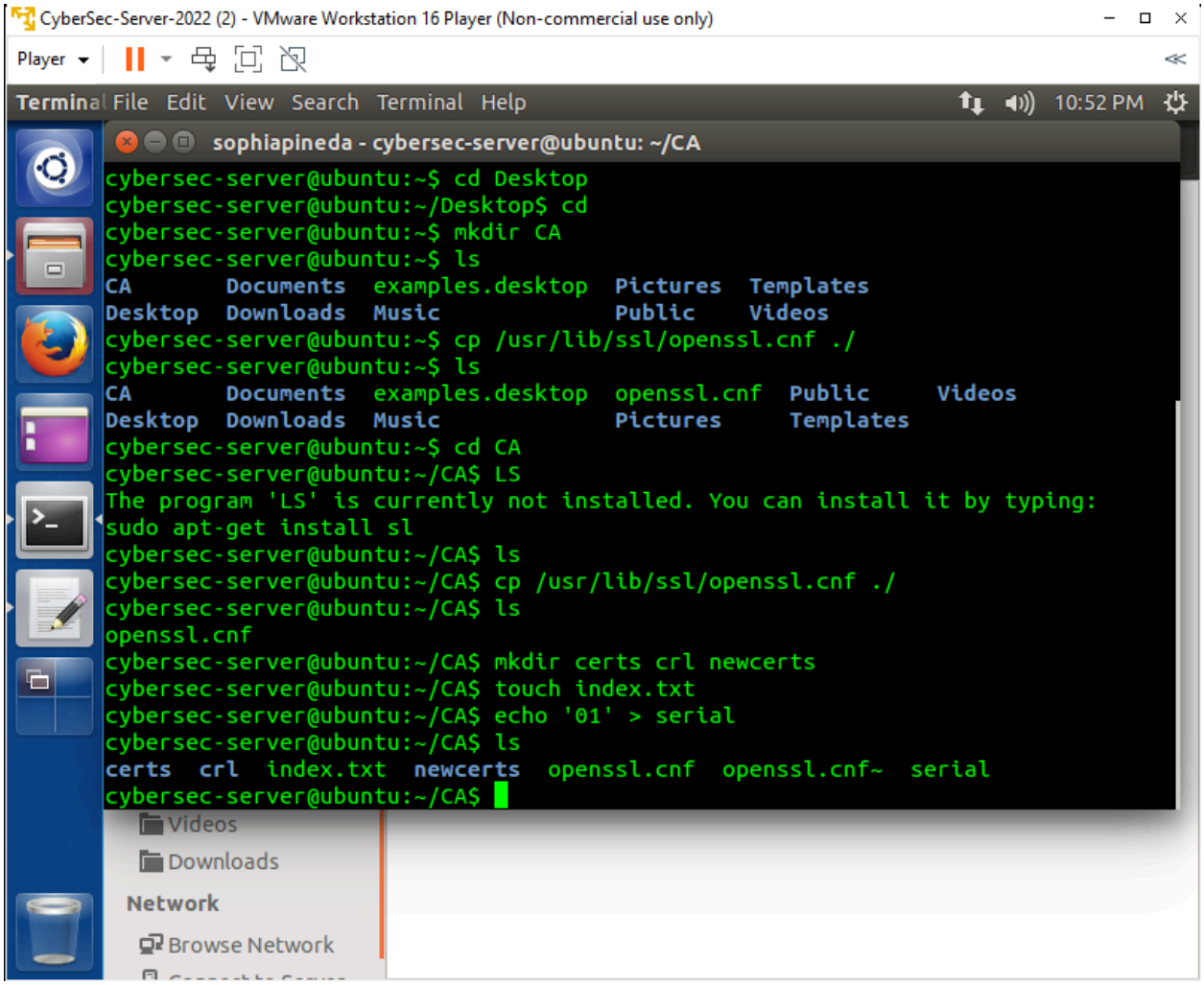
1. Command to generate self-signed certificate for the CA

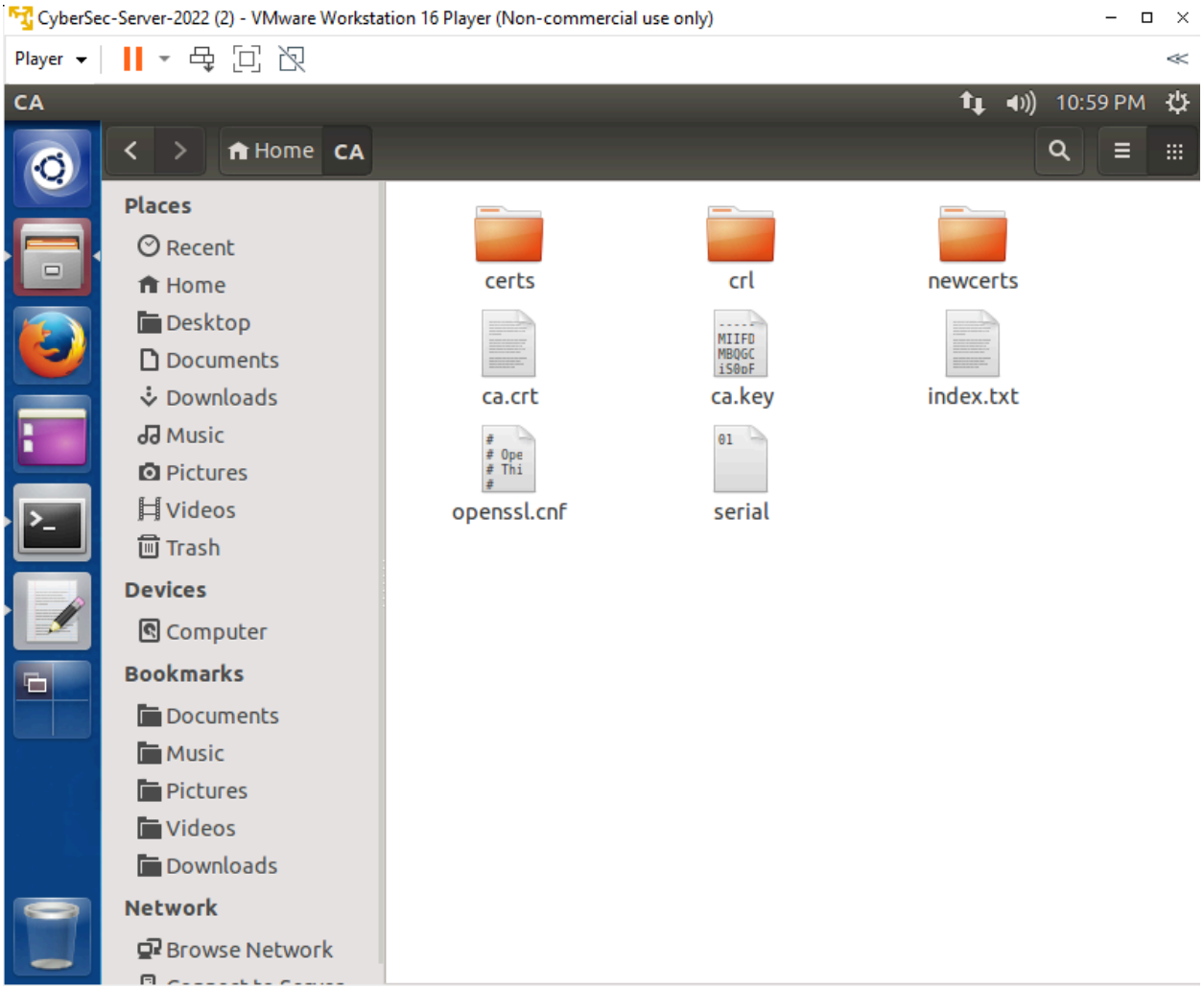


2. Info given for generating the self-signed certificate



3. List of files created


CyberSec-Server-2022 (2) - VMware Workstation 16 Player (Non-commercial use only)
Player
Terminal File Edit View Search Terminal Help
10:52 PM
sophiapineda - cybersec-server@ubuntu: ~/CA
cybersec-server@ubuntu:~$ cd Desktop
cybersec-server@ubuntu:~/Desktop$ cd
cybersec-server@ubuntu:~$ mkdir CA
cybersec-server@ubuntu:~$ ls
CA Documents examples.desktop Pictures Templates
Desktop Downloads Music Public Videos
cybersec-server@ubuntu:~$ cp /usr/lib/ssl/openssl.cnf ./
cybersec-server@ubuntu:~$ ls
CA Documents examples.desktop openssl.cnf Public Videos
Desktop Downloads Music Pictures Templates
cybersec-server@ubuntu:~$ cd CA
cybersec-server@ubuntu:~/CA$ LS
The program 'LS' is currently not installed. You can install it by typing:
sudo apt-get install sl
cybersec-server@ubuntu:~/CA$ ls
cybersec-server@ubuntu:~/CA$ cp /usr/lib/ssl/openssl.cnf ./
cybersec-server@ubuntu:~/CA$ ls
openssl.cnf
cybersec-server@ubuntu:~/CA$ mkdir certs crl newcerts
cybersec-server@ubuntu:~/CA$ touch index.txt
cybersec-server@ubuntu:~/CA$ echo '01' > serial
cybersec-server@ubuntu:~/CA$ ls
certs crl index.txt newcerts openssl.cnf openssl.cnf~ serial
cybersec-server@ubuntu:~/CA$
Videos
Downloads
Network
Browse Network



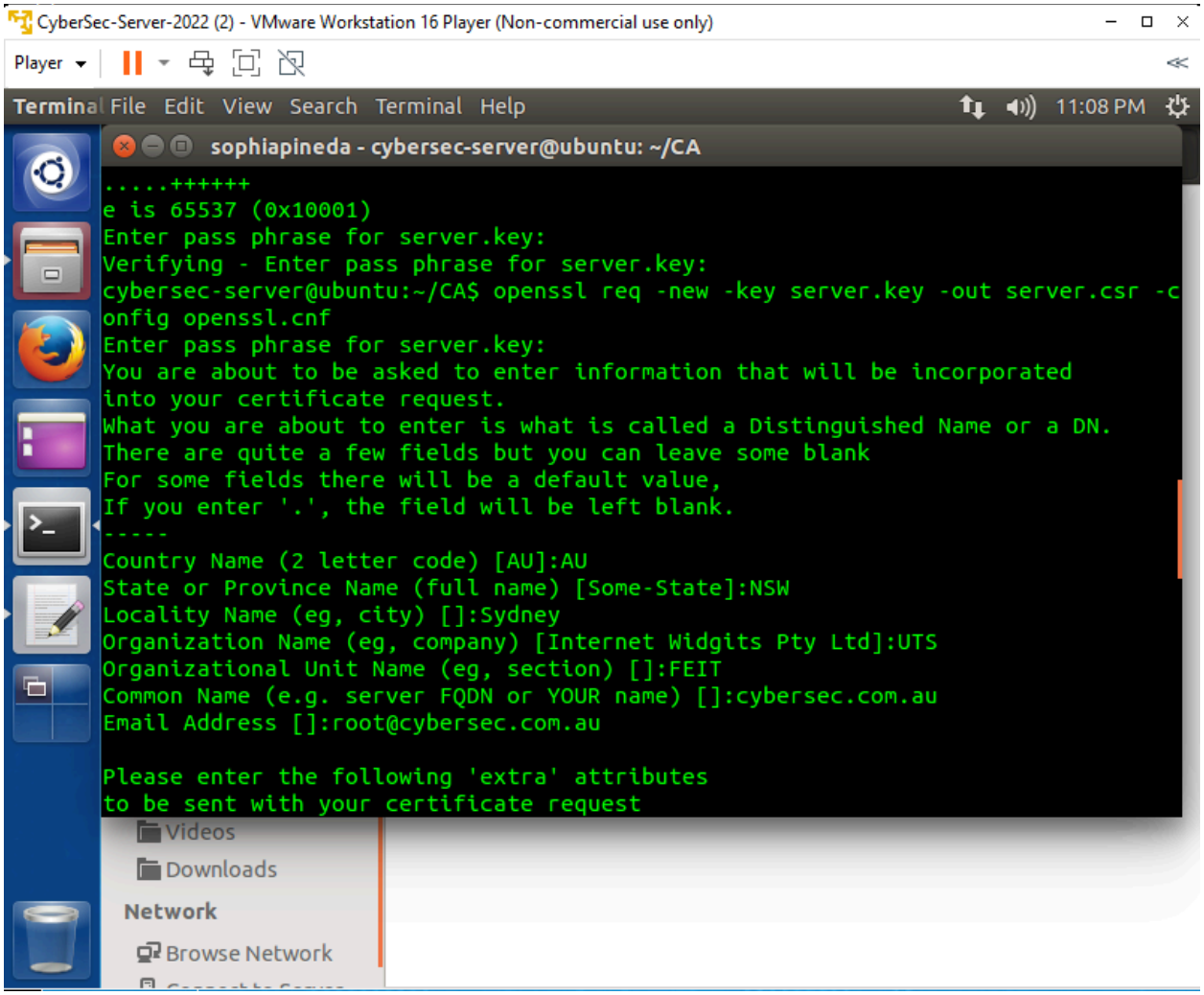
TASK 3: CREATE A CERTIFICATE FOR CYBERSEC.COM.AU

STEPS:

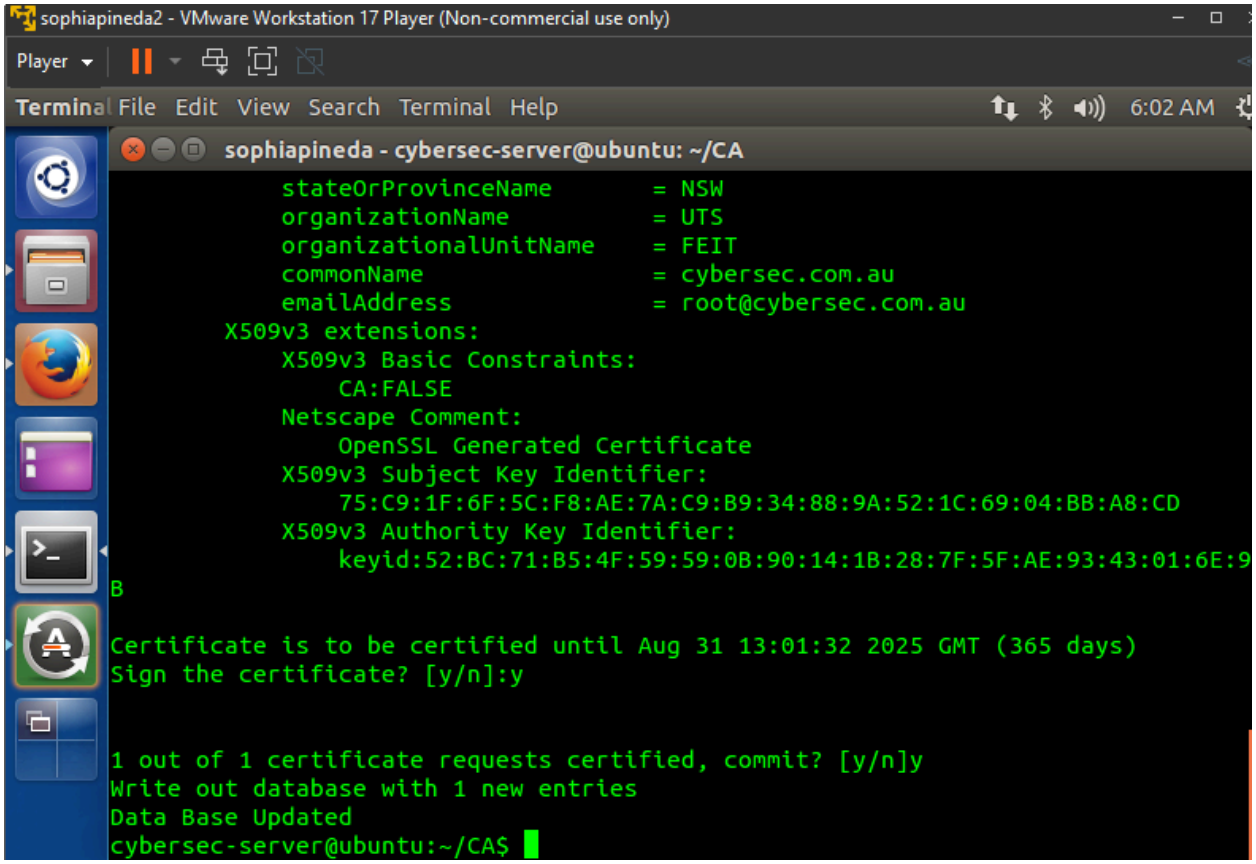
- Generate public/private key pair, use command
- Generate certificate signing request (company's public key)
- Turn certificate signing request into an X509 certificate

SCREENSHOTS

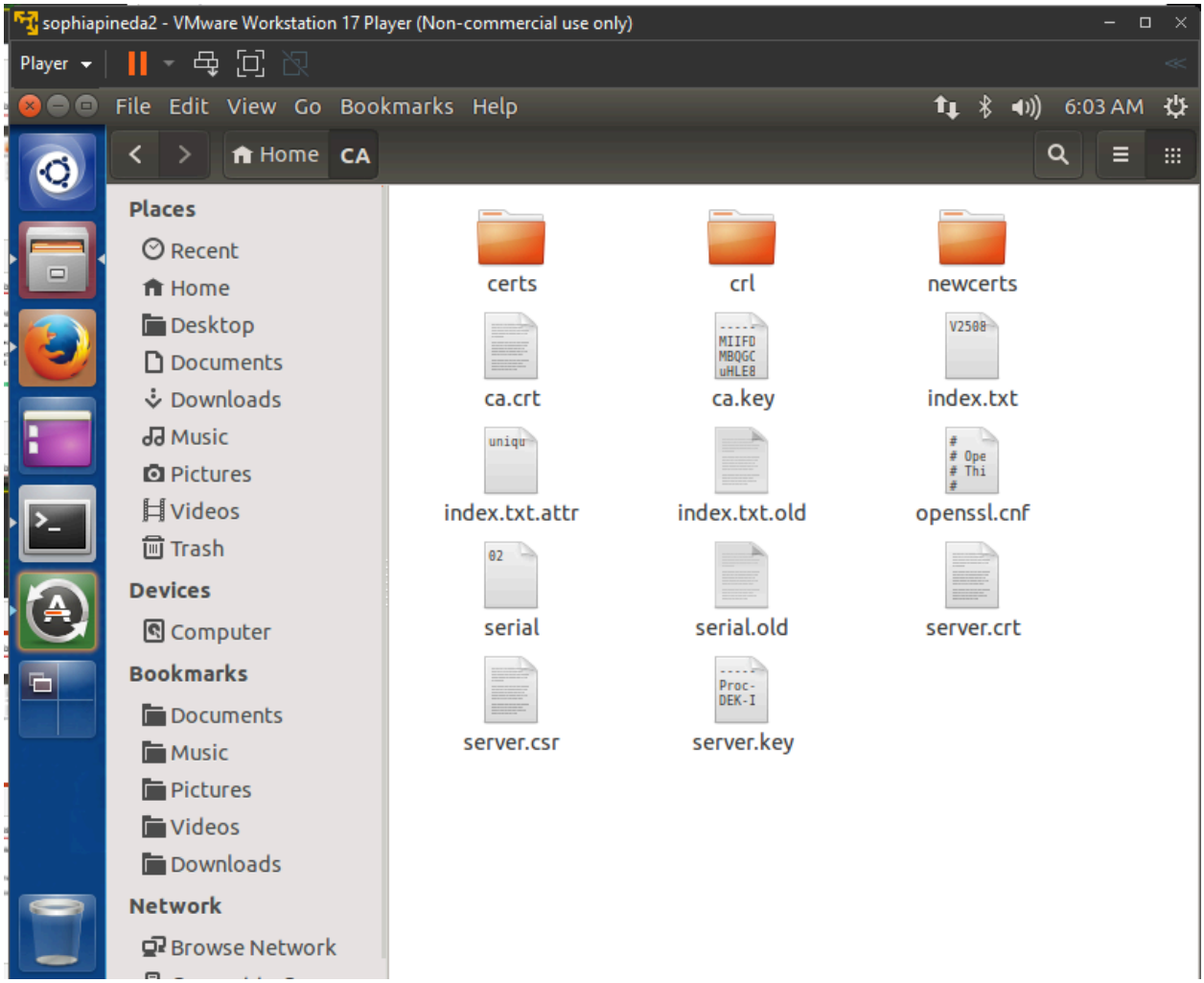
1. Command used to generate CSR



2. Info given when generating CSR



3. Files created



TASK 4: USE PKI FOR WEBSITES

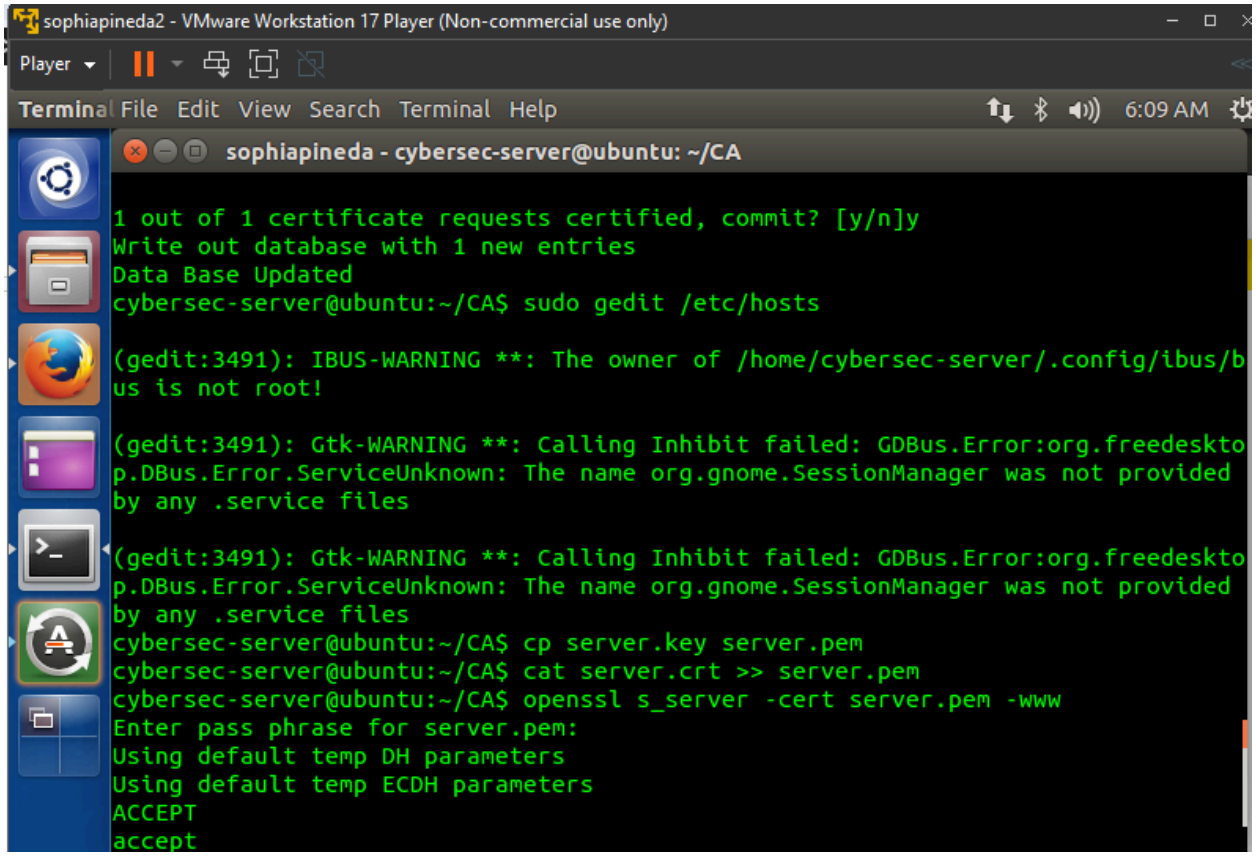
- How public-key certificates are used by websites to secure web browsing

STEPS:

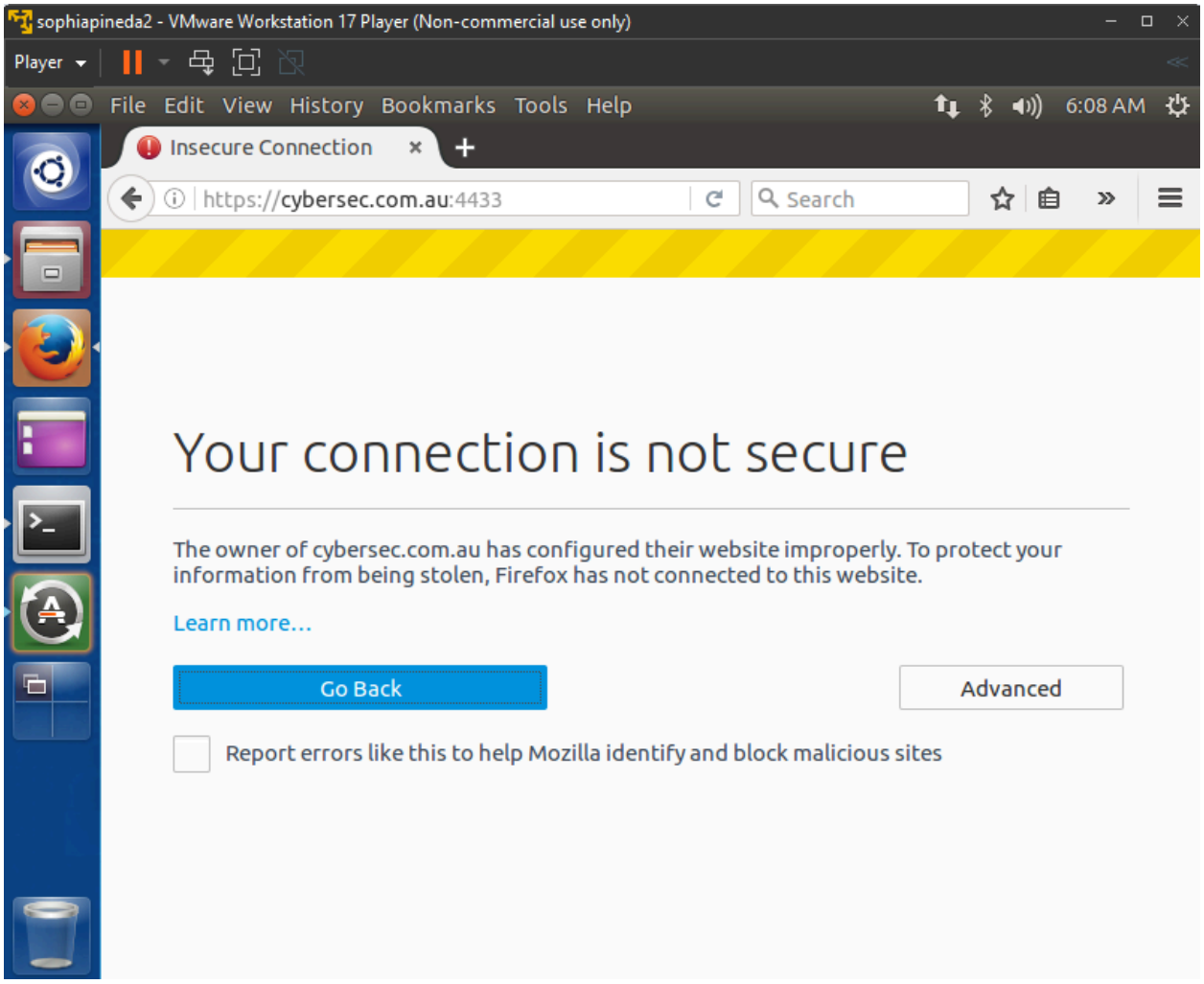
- Make domain name add to local host files
- Restart system
- Launch web server with certificate generated from previous task
- Certificate + secret key
- Manually add our CA's certificate to firefox's browser

SCREENSHOTS:

1. Launching the web server using "server.pem"



2. Screenshot of certificate manager showing the generated certificate for cybersec.com.au



3. Website on web browser after the addition of the certificate

sophiapineda2 - VMware Workstation 17 Player (Non-commercial use only)

Player

Mozilla Firefox 6:12 AM

https://cyb...om.au:4433/ Preferences

https://cybersec.com.au:4433 Search

```
Ciphers common between both SSL end points:
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES256-SHA
ECDHE-ECDSA-AES128-SHA     ECDHE-RSA-AES128-SHA       ECDHE-RSA-AES256-SHA
DHE-RSA-AES128-SHA         DHE-RSA-AES256-SHA         AES128-SHA
AES256-SHA                 DES-CBC3-SHA
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES128-GCM-SHA256
    Session-ID:
    Session-ID-ctx: 01000000
    Master-Key: 6257C7840D2DAB5F68F3389B266044BF926BC2D59724317B53D509AC23AE900CEC0C8B483192AAC920BA9F
    Key-Arg   : None
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    Start Time: 1725109925
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
   0 items in the session cache
   0 client connects (SSL_connect())
   0 client renegotiates (SSL_connect())
   0 client connects that finished
   3 server accepts (SSL_accept())
   0 server renegotiates (SSL_accept())
   3 server accepts that finished
   0 session cache hits
   0 session cache misses
   0 session cache timeouts
   0 callback cache hits
   0 cache full overflows (128 allowed)
```