SECURITY KEY-BASED NETWORK ACCESS CONTROL IN CLOUD INSTANCES

MENTOR:

- MARUTHI INUKONDA

TEAM MEMBERS:

- PRANEETH NAYAK (CS20BTECH11025)
- KETAN SABNE(CS20BTECH11043)

PROBLEM STATEMENT :

- To provide a secure key-based network authentication for headless Linux systems in cloud instances that use network access control (NAC) and prevent credential leakage via virtual machine (VM) disk images.
- This involves implementing certificate-based authentication in netplan and utilizing encrypted file systems or disk volumes to ensure zero-leakage of authentication credentials.
- The goal is to obtain fine-grained identity information (user, instance/device) from network logs to improve security in cloud instances, especially as they are increasingly being used for cybercrime hosting services.

MOTIVATION :

- The aim of this project is to provide a secure key-based network authentication for headless Linux systems in cloud instances, using certificates and preventing any leakage through encrypted file systems or disk volumes. Also, The current authentication process for headless Linux systems using netplan configuration files is prone to credential leakage, which can compromise the security of cloud instances.
- By providing a secure and zero-leakage key-based network authentication using certificates and encrypted file-systems/disk-volumes,we can enhance the security of NAC in cloud instances and prevent unauthorized access by cybercriminals.

APPROACH :

Tools

- Setting up a RADIUS server:We are going to use a RADIUS server such as FreeRADIUS or Microsoft Network Policy Server (NPS) to authenticate network connections. This involves configuring network clients to use it.
- Configuring netplan for cert based authentication: Netplan is a network configuration tool used in Ubuntu. Because of this no need of password and hashes in config files.
- Using encrypted file systems/volumes: To prevent any leakage of authentication credentials via VM,we will use encrypted file systems or volumes.We will use tool LUKS (Linux Unified Key Setup) to create an encrypted volume.
- Setting up VM: VM will be provided by you with access.

APPROACH (CONTD):

How we will do:

1. I have already written in timelines our about our plans that how we are going to proceed weekly.
2. The approach to achieve this could be to implement a certificate-based authentication mechanism in the netplan configuration file for cloud instances. This would involve generating a unique key pair for each instance, with the public key being added to the RADIUS server for authentication. The private key would be securely stored in an encrypted file system or disk volume to prevent any leakage.

UNDERSTANDING NAC,IAM AND CERT. BASED AUTHENTICATION.

- NAC: Enforces policies on devices connecting to a network
- IAM: Manages digital identities and access rights
- Cert-based auth: Uses digital certificates to verify identity
- Certificates issued by trusted authority
- Provides secure communication over the internet
- ***Conclusion***: Considered more secure than traditional password-based auth.

IMPLEMENTING ENCRYPTED FILE-SYSTEMS/DISK-VOLUMES TO PREVENT NETWORK ACCESS CREDENTIALS FROM BEING LEAK.

- Install encryption software such as LUKS.
- Create an encrypted file-system or disk volume and set a password or passphrase for the encryption key
- Mount the encrypted file-system or disk volume using the mount command
- Set up automatic decryption to avoid manual entry of the encryption key
- Secure the encryption key by storing it in a secure location and limiting access to authorized personnel

IMPLEMENTING ENCRYPTED FILE-SYSTEMS/DISK-VOLUMES TO PREVENT NETWORK ACCESS CREDENTIALS FROM BEING LEAK.

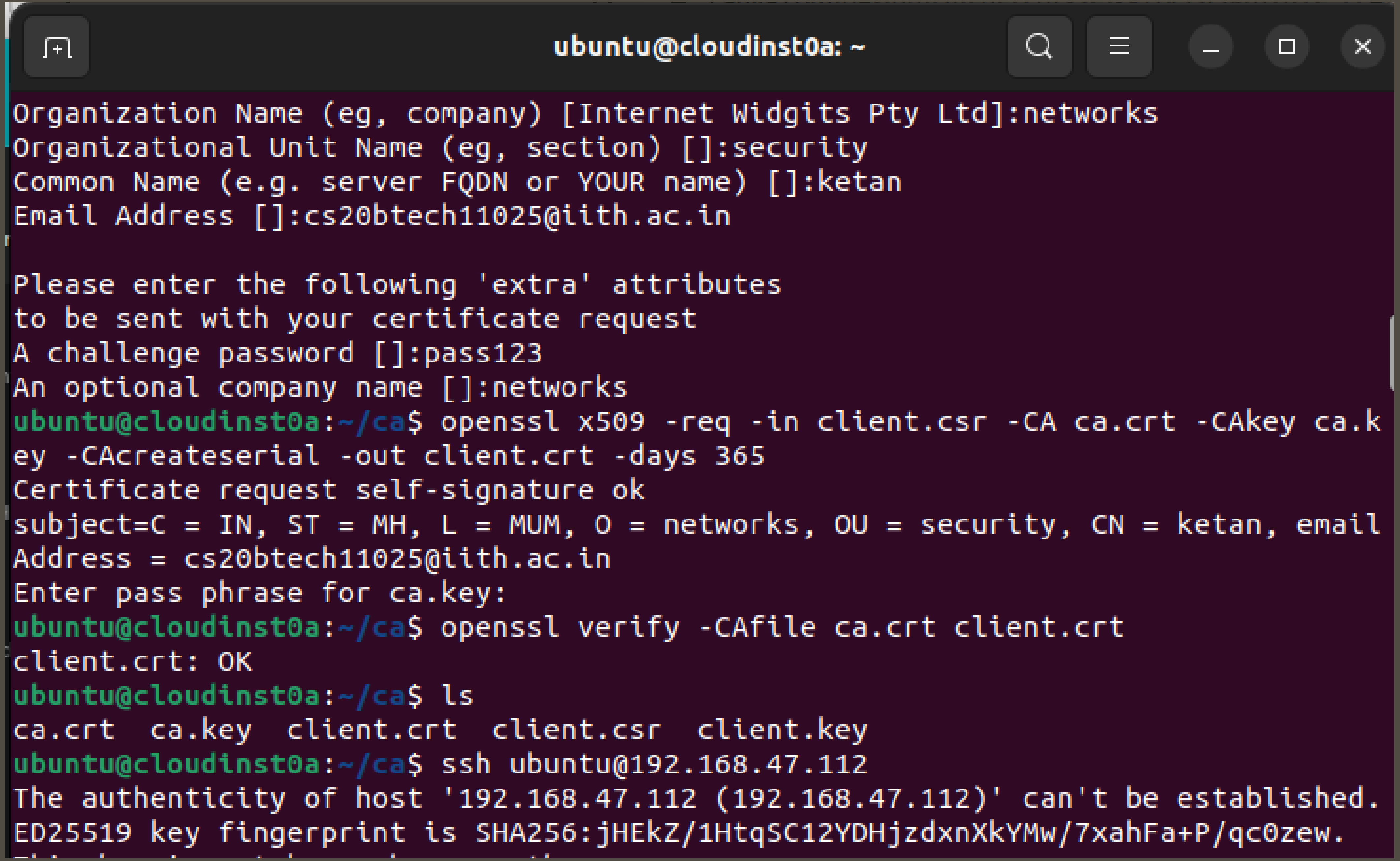
- for creating an encrypted file system on a headless Linux system using LUKS:
- Install the cryptsetup package: sudo apt-get install cryptsetup (for Ubuntu/Debian-based systems)
- Create an encrypted volume: sudo cryptsetup luksFormat /dev/sdb1.
- Open the encrypted volume: sudo cryptsetup luksOpen /dev/sdb1 my_encrypted_volume.

IMPLEMENTING ENCRYPTED FILE-SYSTEMS/DISK-VOLUMES TO PREVENT NETWORK ACCESS CREDENTIALS FROM BEING LEAK.

- Create a file system: sudo mkfs.ext4 /dev/mapper/my_encrypted_volume.
- Mount the file system: sudo mount /dev/mapper/my_encrypted_volume /mnt/my_encrypted_volume.
- Unmount the file system: sudo umount /mnt/my_encrypted_volume.
- Close the encrypted volume: sudo cryptsetup luksClose my_encrypted_volume

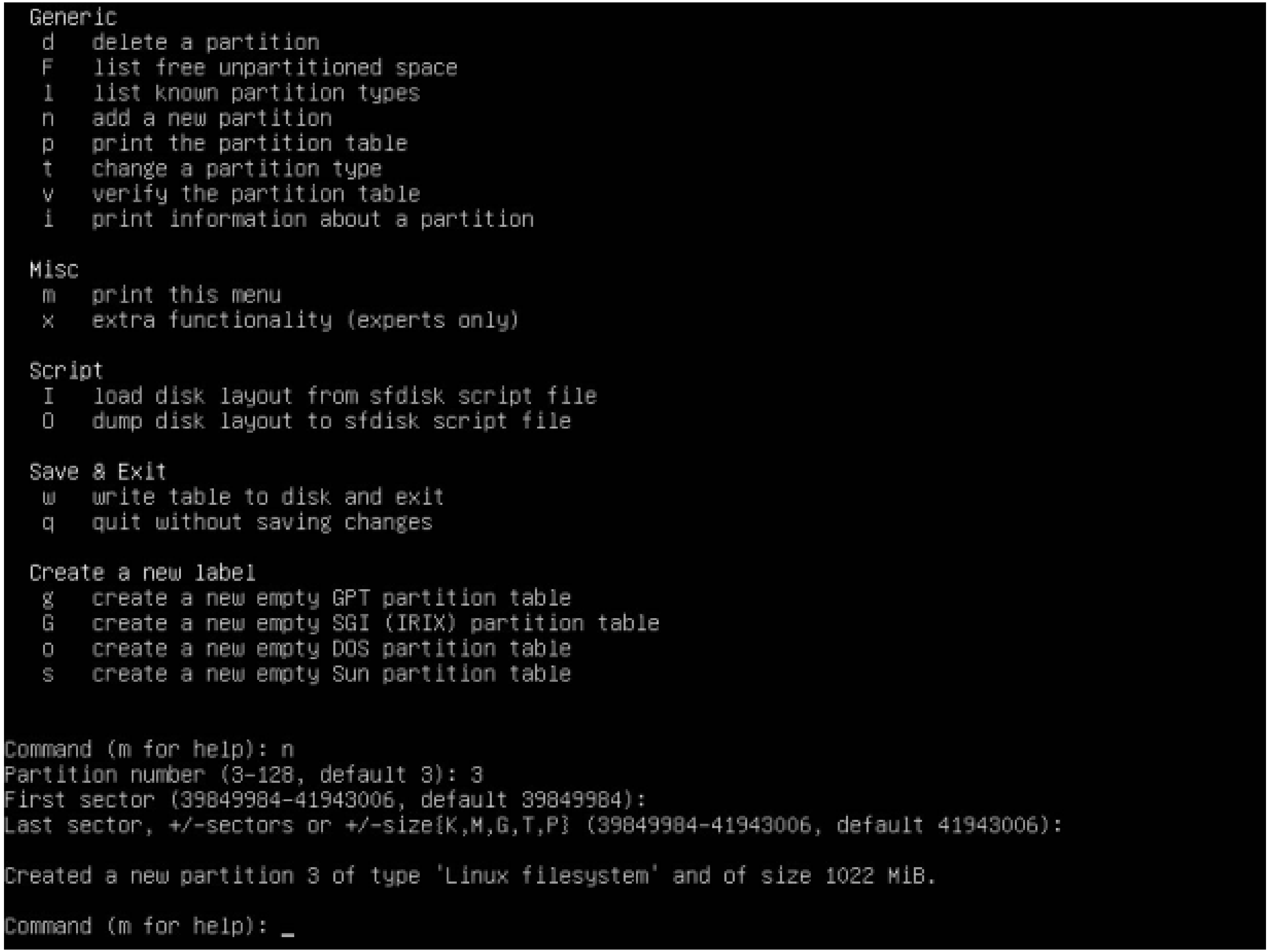
CREATING CERTFICATES:

- Certificate Authority and client certificates generated and signed in vm (cloudinst0a).



```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:networks
Organizational Unit Name (eg, section) []:security
Common Name (e.g. server FQDN or YOUR name) []:ketan
Email Address []:cs20btech11025@iith.ac.in

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:pass123
An optional company name []:networks
ubuntu@cloudinst0a:~/ca$ openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.k
ey -CAcreateserial -out client.crt -days 365
Certificate request self-signature ok
subject=C = IN, ST = MH, L = MUM, O = networks, OU = security, CN = ketan, email
Address = cs20btech11025@iith.ac.in
Enter pass phrase for ca.key:
ubuntu@cloudinst0a:~/ca$ openssl verify -CAfile ca.crt client.crt
client.crt: OK
ubuntu@cloudinst0a:~/ca$ ls
ca.crt  ca.key  client.crt  client.csr  client.key
ubuntu@cloudinst0a:~/ca$ ssh ubuntu@192.168.47.112
The authenticity of host '192.168.47.112 (192.168.47.112)' can't be established.
ED25519 key fingerprint is SHA256:jHEkZ/1HtqSC12YDHjzdxnXkYMw/7xahFa+P/qc0zew.
```

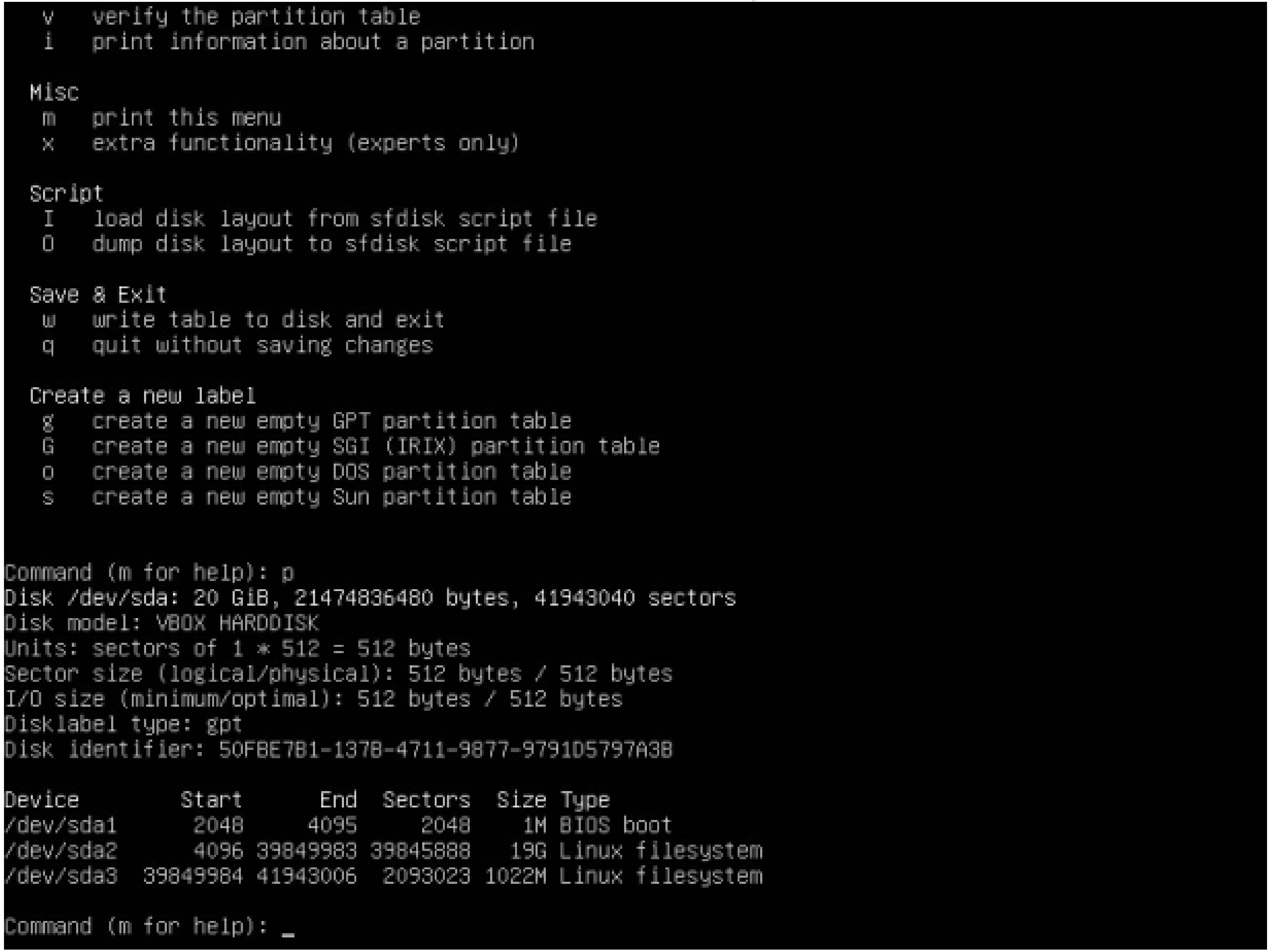
IMPLEMENTATION



```
  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help): n
Partition number (3-128, default 3): 3
First sector (39849984-41943006, default 39849984):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (39849984-41943006, default 41943006):

Created a new partition 3 of type 'Linux filesystem' and of size 1022 MiB.

Command (m for help): _
```

- Encrypting the keys stored on the file-system (/secure) a partition is created using sudo fdisk /dev/sda command , below Screenshot is step for creating new partition, for me new partition name created is /dev/sda3.

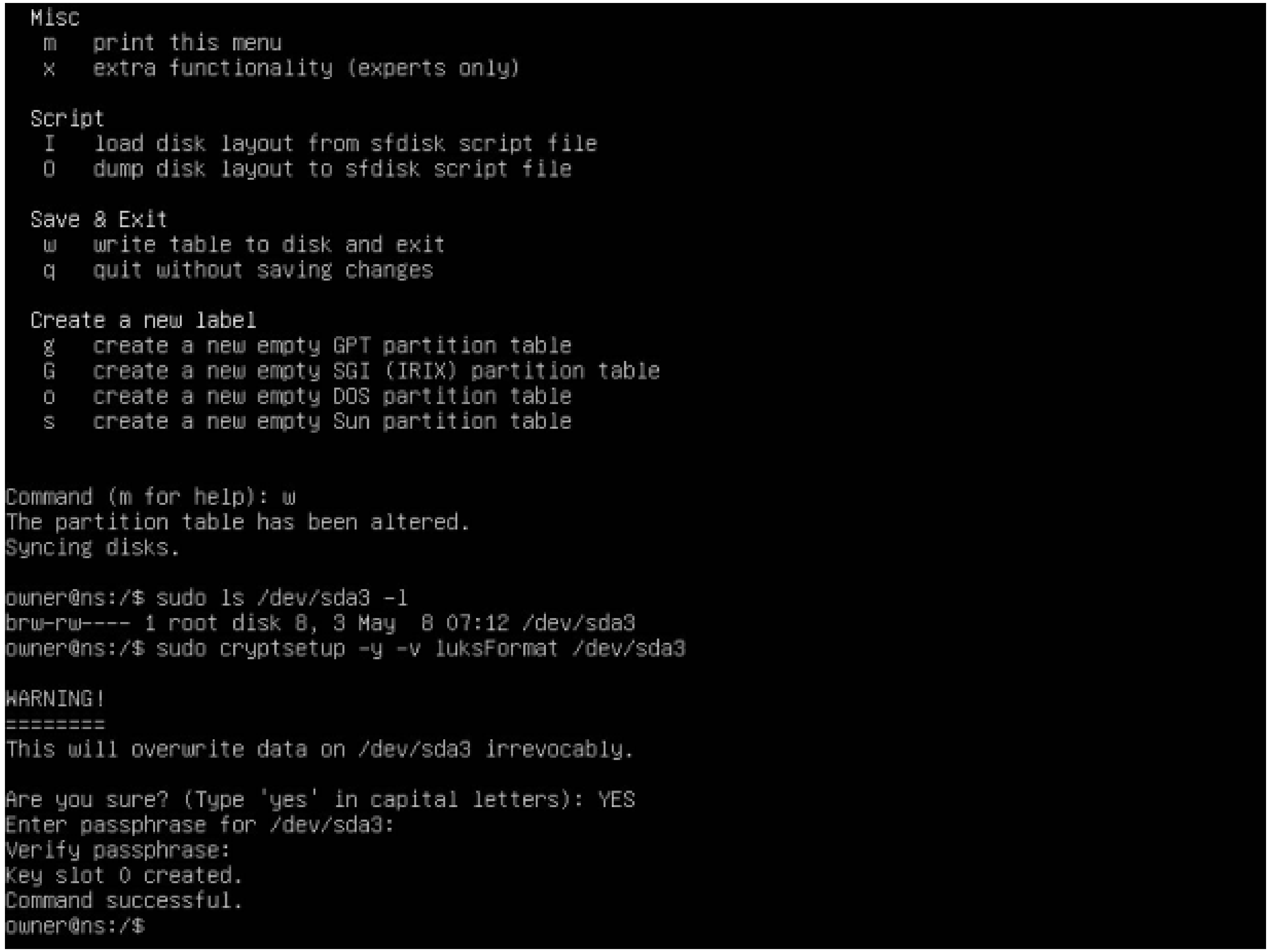
IMPLEMENTATION



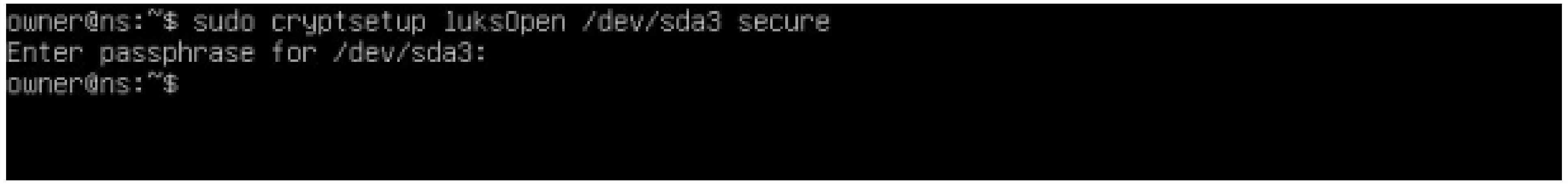
```
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help): p
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 50FBE7B1-137B-4711-9877-9791D5797A38

Device        Start      End  Sectors  Size Type
/dev/sda1      2048     4095     2048    1M BIOS boot
/dev/sda2      4096 39849983 39845888   19G Linux filesystem
/dev/sda3  39849984 41943006  2093023 1022M Linux filesystem

Command (m for help): _
```

- checked whether partition created or not.

IMPLEMENTATION



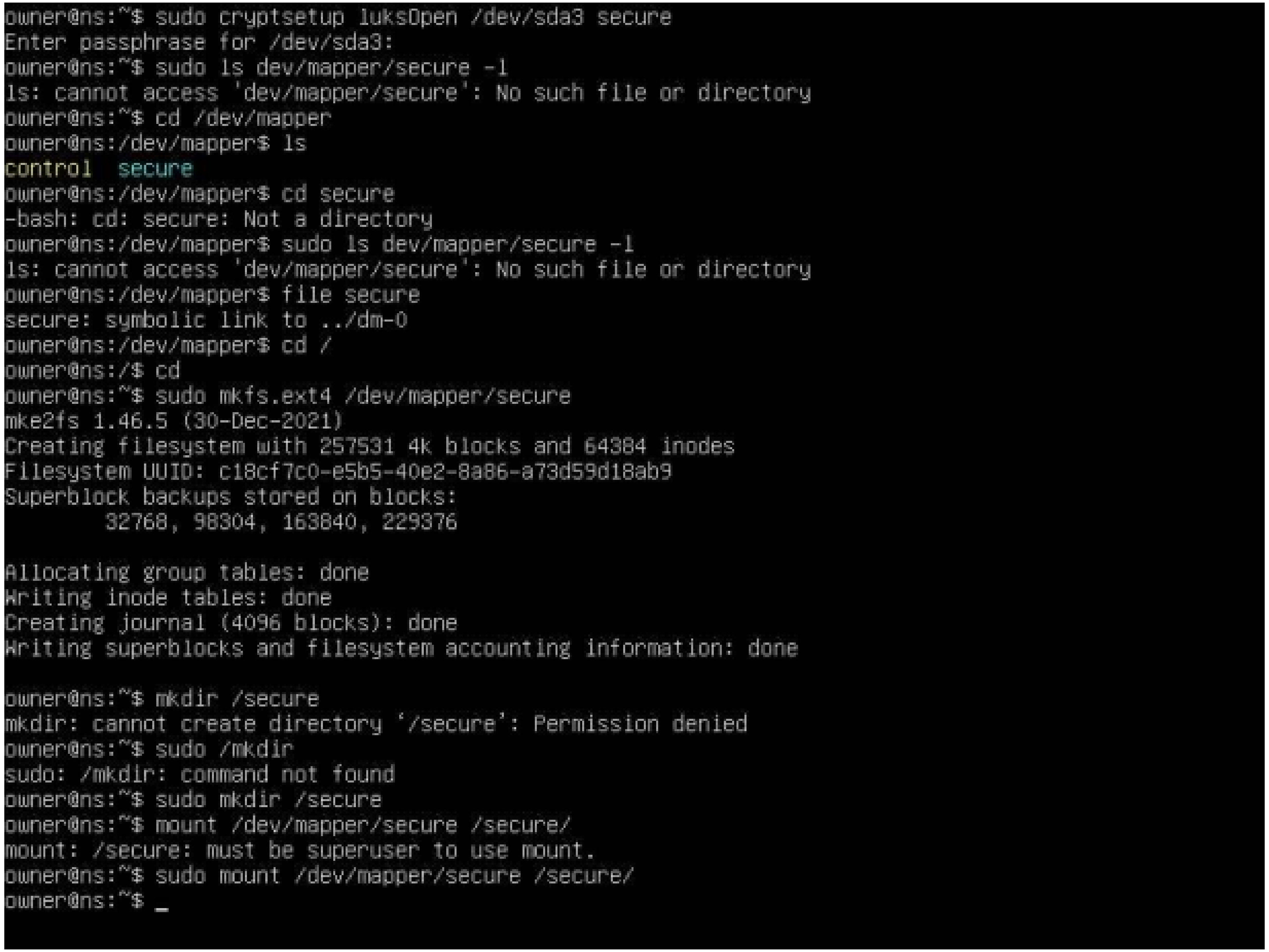
```
  Misc
   m   print this menu
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help): w
The partition table has been altered.
Syncing disks.

owner@ns:/$ sudo ls /dev/sda3 -l
brw-rw---- 1 root disk 8, 3 May  8 07:12 /dev/sda3
owner@ns:/$ sudo cryptsetup -y -v luksFormat /dev/sda3

WARNING!
========
This will overwrite data on /dev/sda3 irrevocably.

Are you sure? (Type 'yes' in capital letters): YES
Enter passphrase for /dev/sda3:
Verify passphrase:
Key slot 0 created.
Command successful.
owner@ns:/$
```



```
owner@ns:~$ sudo cryptsetup luksOpen /dev/sda3 secure
Enter passphrase for /dev/sda3:
owner@ns:~$
```

- Setting up my LUKS partition with cryptsetup command.
- Then, open the partition and create a mapping in /dev/mapper using below command.
- Cmd : sudo cryptsetup luksOpen /dev/sda3 secure.

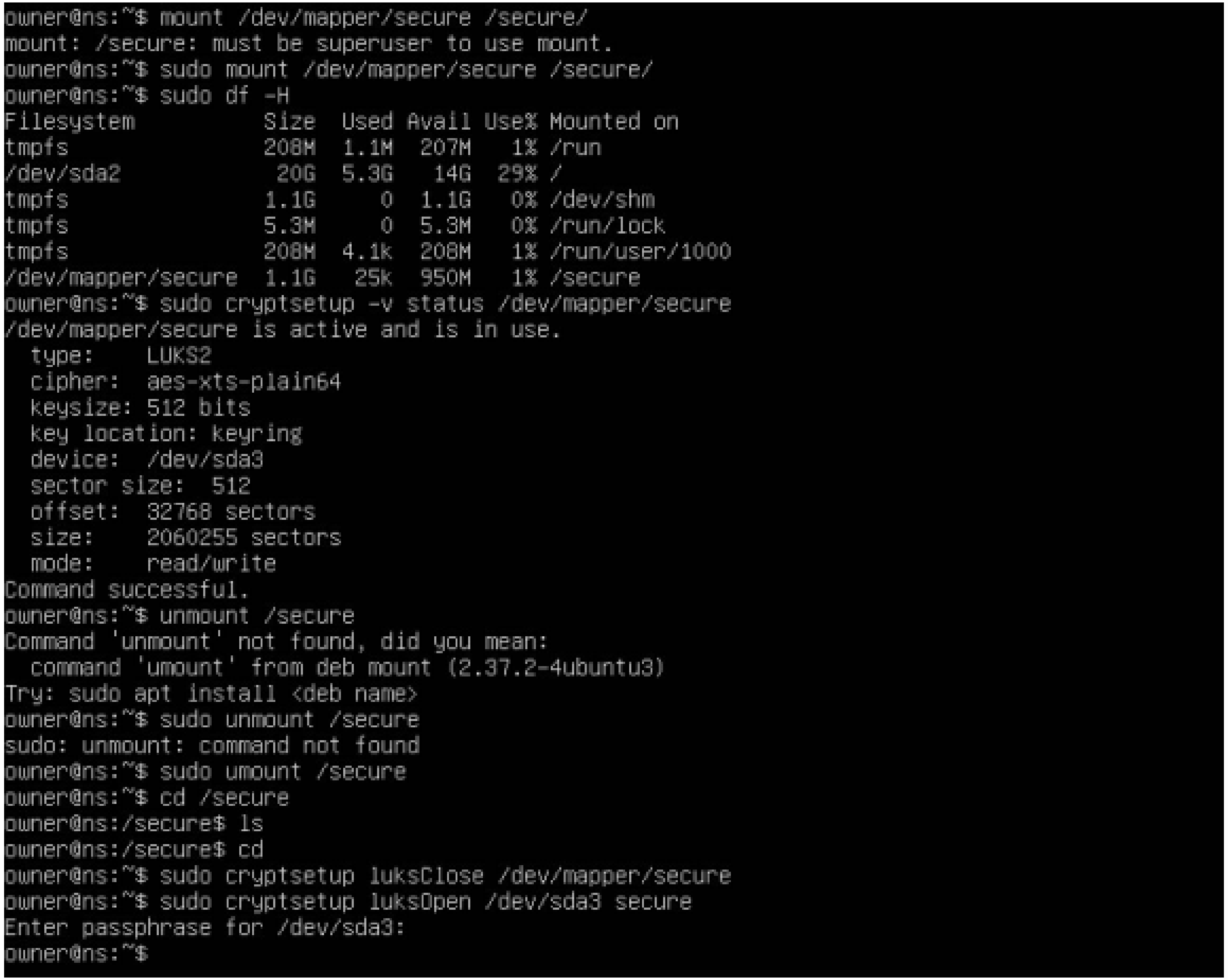
IMPLEMENTATION



```
owner@ns:~$ sudo cryptsetup luksOpen /dev/sda3 secure
Enter passphrase for /dev/sda3:
owner@ns:~$ sudo ls dev/mapper/secure -l
ls: cannot access 'dev/mapper/secure': No such file or directory
owner@ns:~$ cd /dev/mapper
owner@ns:/dev/mapper$ ls
control  secure
owner@ns:/dev/mapper$ cd secure
-bash: cd: secure: Not a directory
owner@ns:/dev/mapper$ sudo ls dev/mapper/secure -l
ls: cannot access 'dev/mapper/secure': No such file or directory
owner@ns:/dev/mapper$ file secure
secure: symbolic link to ../dm-0
owner@ns:/dev/mapper$ cd /
owner@ns:/$ cd
owner@ns:~$ sudo mkfs.ext4 /dev/mapper/secure
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 257531 4k blocks and 64384 inodes
Filesystem UUID: c18cf7c0-e5b5-40e2-8a86-a73d59d18ab9
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

owner@ns:~$ mkdir /secure
mkdir: cannot create directory '/secure': Permission denied
owner@ns:~$ sudo /mkdir
sudo: /mkdir: command not found
owner@ns:~$ sudo mkdir /secure
owner@ns:~$ mount /dev/mapper/secure /secure/
mount: /secure: must be superuser to use mount.
owner@ns:~$ sudo mount /dev/mapper/secure /secure/
owner@ns:~$ _
```

- Check that our setup name (secure) is mapped correctly or not.
- Create a file system with mkfs.ext4 command and mounting a new file system which will be used in storing certificates.

IMPLEMENTATION

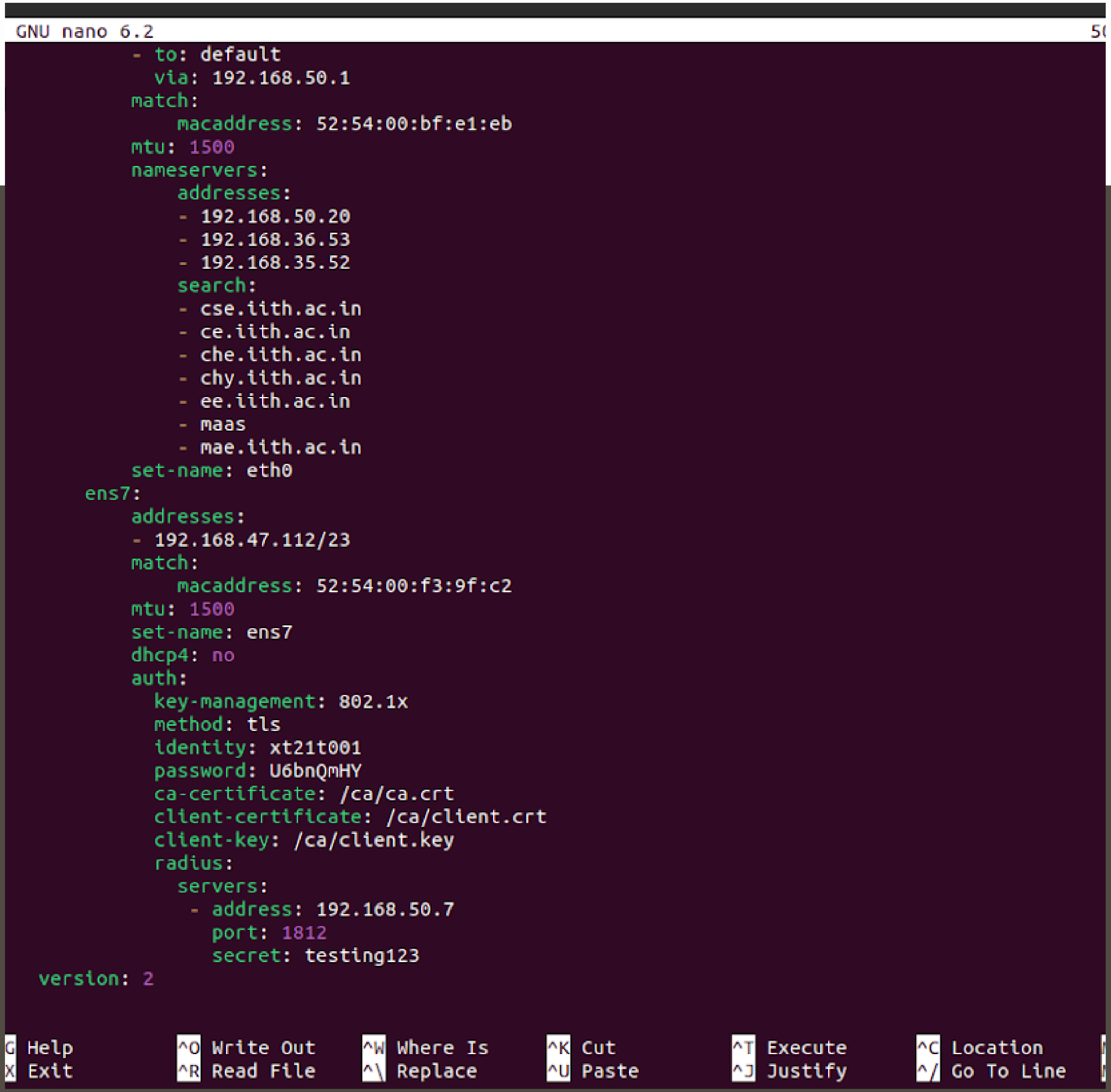


```
owner@ns:~$ mount /dev/mapper/secure /secure/
mount: /secure: must be superuser to use mount.
owner@ns:~$ sudo mount /dev/mapper/secure /secure/
owner@ns:~$ sudo df -H
Filesystem          Size  Used Avail Use% Mounted on
tmpfs               208M  1.1M  207M   1% /run
/dev/sda2            20G  5.3G   14G  29% /
tmpfs               1.1G     0  1.1G   0% /dev/shm
tmpfs               5.3M     0  5.3M   0% /run/lock
tmpfs               208M  4.1k  208M   1% /run/user/1000
/dev/mapper/secure  1.1G   25k  950M   1% /secure
owner@ns:~$ sudo cryptsetup -v status /dev/mapper/secure
/dev/mapper/secure is active and is in use.
  type:    LUKS2
  cipher:  aes-xts-plain64
  keysize: 512 bits
  key location: keyring
  device:  /dev/sda3
  sector size:  512
  offset:  32768 sectors
  size:    2060255 sectors
  mode:    read/write
Command successful.
owner@ns:~$ unmount /secure
Command 'unmount' not found, did you mean:
  command 'umount' from deb mount (2.37.2-4ubuntu3)
Try: sudo apt install <deb name>
owner@ns:~$ sudo unmount /secure
sudo: unmount: command not found
owner@ns:~$ sudo umount /secure
owner@ns:~$ cd /secure
owner@ns:/secure$ ls
owner@ns:/secure$ cd
owner@ns:~$ sudo cryptsetup luksClose /dev/mapper/secure
owner@ns:~$ sudo cryptsetup luksOpen /dev/sda3 secure
Enter passphrase for /dev/sda3:
owner@ns:~$
```

- Check status of our /dev/mapper/sda3 then, it shows all the info used for encryption like LUKS 2, ciphers and key size,etc.
- Unmount ‘secure’ and now whenever we have to open our partition which is created for certificates, we need to input passphrase.

IMPLEMENTATION

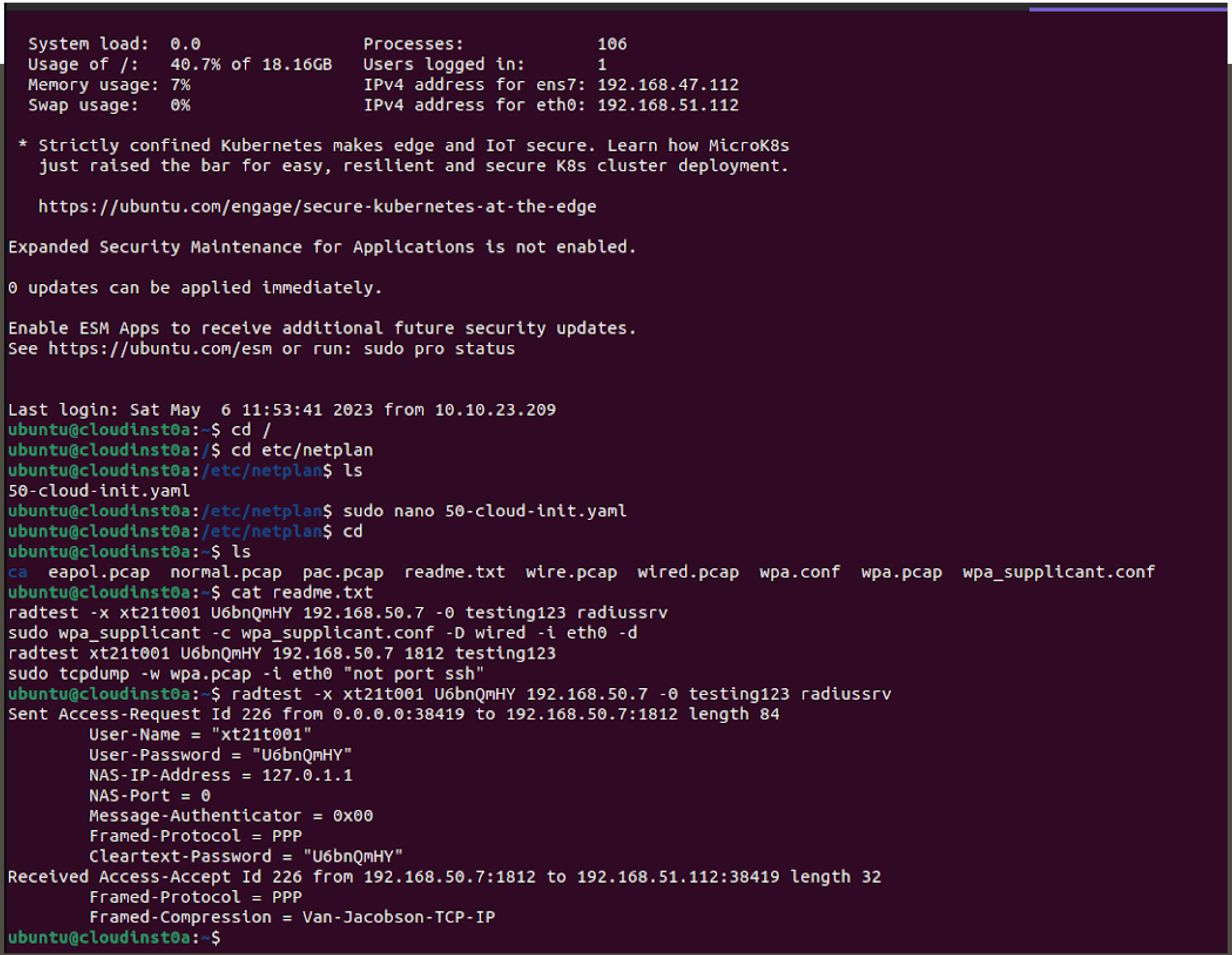
- We are using ens7 interface for authenticating with the use of RADIUS server



```
          - to: default
            via: 192.168.50.1
          match:
              macaddress: 52:54:00:bf:e1:eb
          mtu: 1500
          nameservers:
              addresses:
              - 192.168.50.20
              - 192.168.36.53
              - 192.168.35.52
              search:
              - cse.iith.ac.in
              - ce.iith.ac.in
              - che.iith.ac.in
              - chy.iith.ac.in
              - ee.iith.ac.in
              - maas
              - mae.iith.ac.in
          set-name: eth0
      ens7:
          addresses:
          - 192.168.47.112/23
          match:
              macaddress: 52:54:00:f3:9f:c2
          mtu: 1500
          set-name: ens7
          dhcp4: no
          auth:
            key-management: 802.1x
            method: tls
            identity: xt21t001
            password: U6bnQmHY
            ca-certificate: /ca/ca.crt
            client-certificate: /ca/client.crt
            client-key: /ca/client.key
            radius:
              servers:
               - address: 192.168.50.7
                 port: 1812
                 secret: testing123
  version: 2
```

IMPLEMENTATION

- Radtest, this image shows how we have don radtest using command



```
 System load:  0.0                Processes:             106
 Usage of /:   40.7% of 18.16GB   Users logged in:       1
 Memory usage: 7%                 IPv4 address for ens7: 192.168.47.112
 Swap usage:   0%                 IPv4 address for eth0: 192.168.51.112

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sat May  6 11:53:41 2023 from 10.10.23.209
ubuntu@cloudinst0a:~$ cd /
ubuntu@cloudinst0a:/$ cd etc/netplan
ubuntu@cloudinst0a:/etc/netplan$ ls
50-cloud-init.yaml
ubuntu@cloudinst0a:/etc/netplan$ sudo nano 50-cloud-init.yaml
ubuntu@cloudinst0a:/etc/netplan$ cd
ubuntu@cloudinst0a:~$ ls
ca  eapol.pcap  normal.pcap  pac.pcap  readme.txt  wire.pcap  wired.pcap  wpa.conf  wpa.pcap  wpa_supplicant.conf
ubuntu@cloudinst0a:~$ cat readme.txt
radtest -x xt21t001 U6bnQmHY 192.168.50.7 -0 testing123 radiussrv
sudo wpa_supplicant -c wpa_supplicant.conf -D wired -i eth0 -d
radtest xt21t001 U6bnQmHY 192.168.50.7 1812 testing123
sudo tcpdump -w wpa.pcap -i eth0 "not port ssh"
ubuntu@cloudinst0a:~$ radtest -x xt21t001 U6bnQmHY 192.168.50.7 -0 testing123 radiussrv
Sent Access-Request Id 226 from 0.0.0.0:38419 to 192.168.50.7:1812 length 84
        User-Name = "xt21t001"
        User-Password = "U6bnQmHY"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00
        Framed-Protocol = PPP
        Cleartext-Password = "U6bnQmHY"
Received Access-Accept Id 226 from 192.168.50.7:1812 to 192.168.51.112:38419 length 32
        Framed-Protocol = PPP
        Framed-Compression = Van-Jacobson-TCP-IP
ubuntu@cloudinst0a:~$
```

CONCLUSION:

- Using LUKS, we can encrypt the partition and the certificates are stored in the encrypted partition. So, there wont be any problem as passphrase is required to access certificates.
- By modifying NETPLAN configuration file, the networking on our sytem can be configured.
- Example : Managing interfaces and authentication methods, etc.

THANK YOU