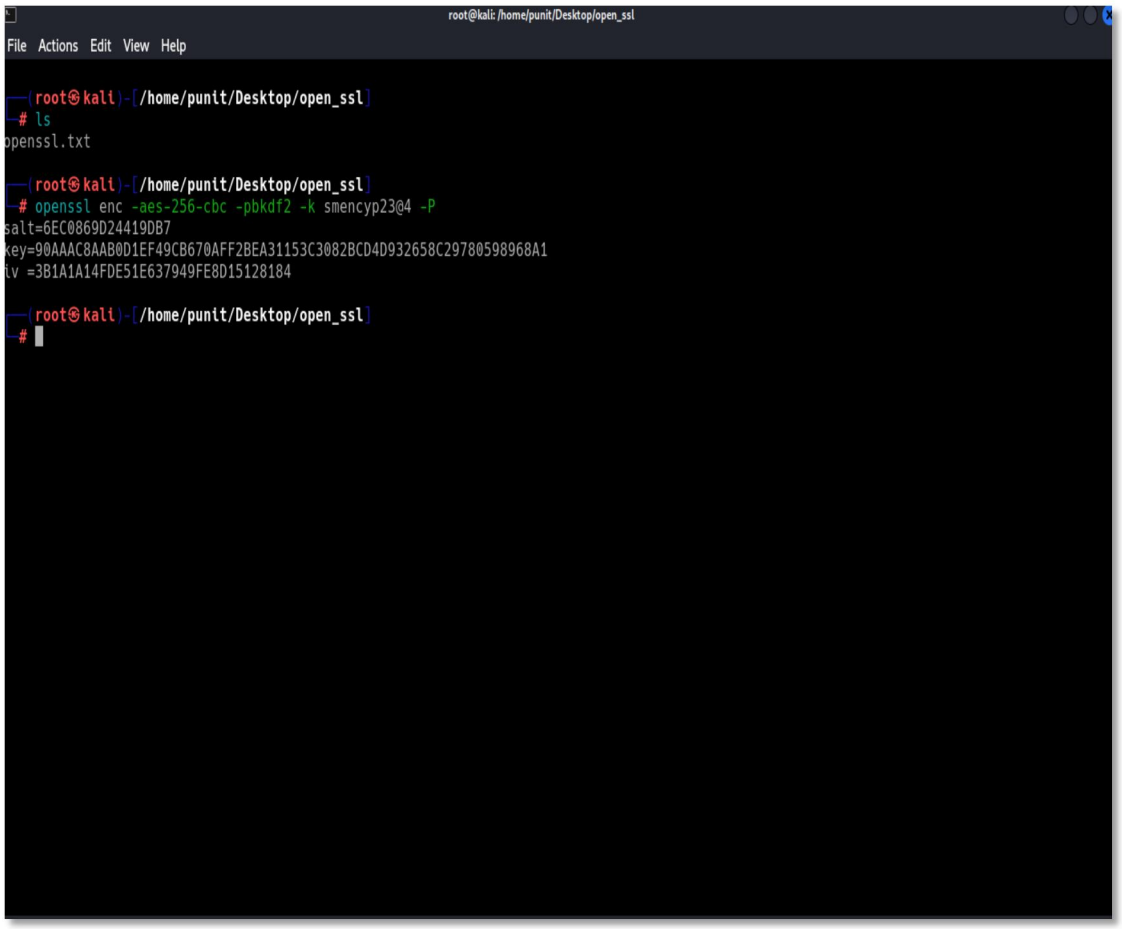
PRACTICAL 9

AIM: Implement encryption and decryption with openssl.

Steps:

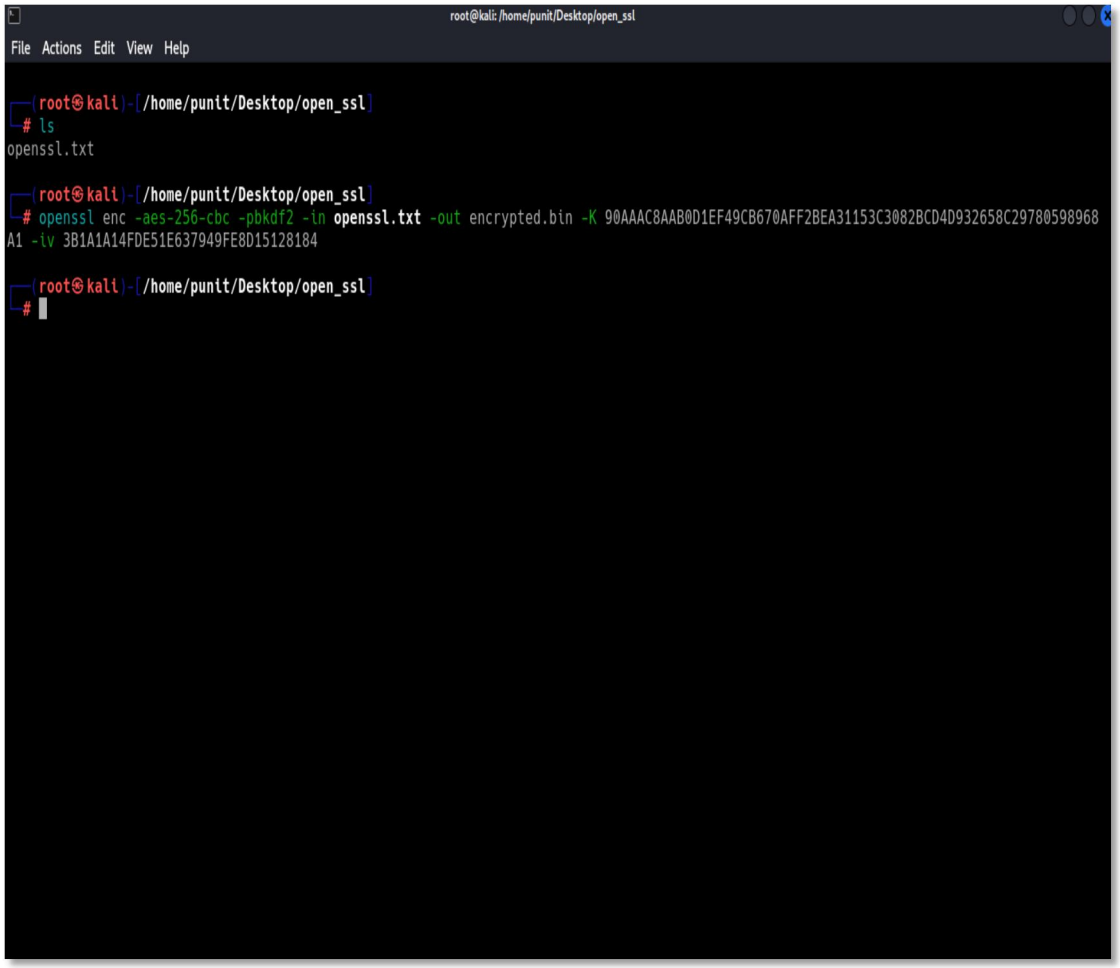
1. Generate an Encryption Key and IV:

openssl enc -aes-256-cbc -pbkdf2 -k smencyp23@4 -P



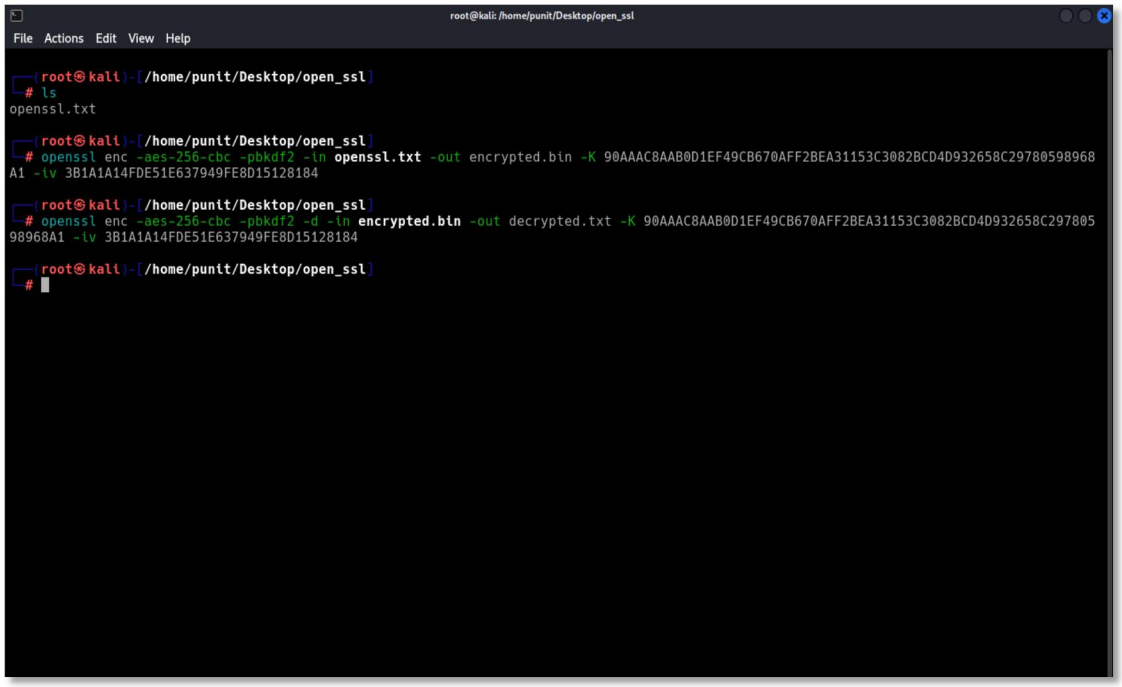
2. Encrypt a File:

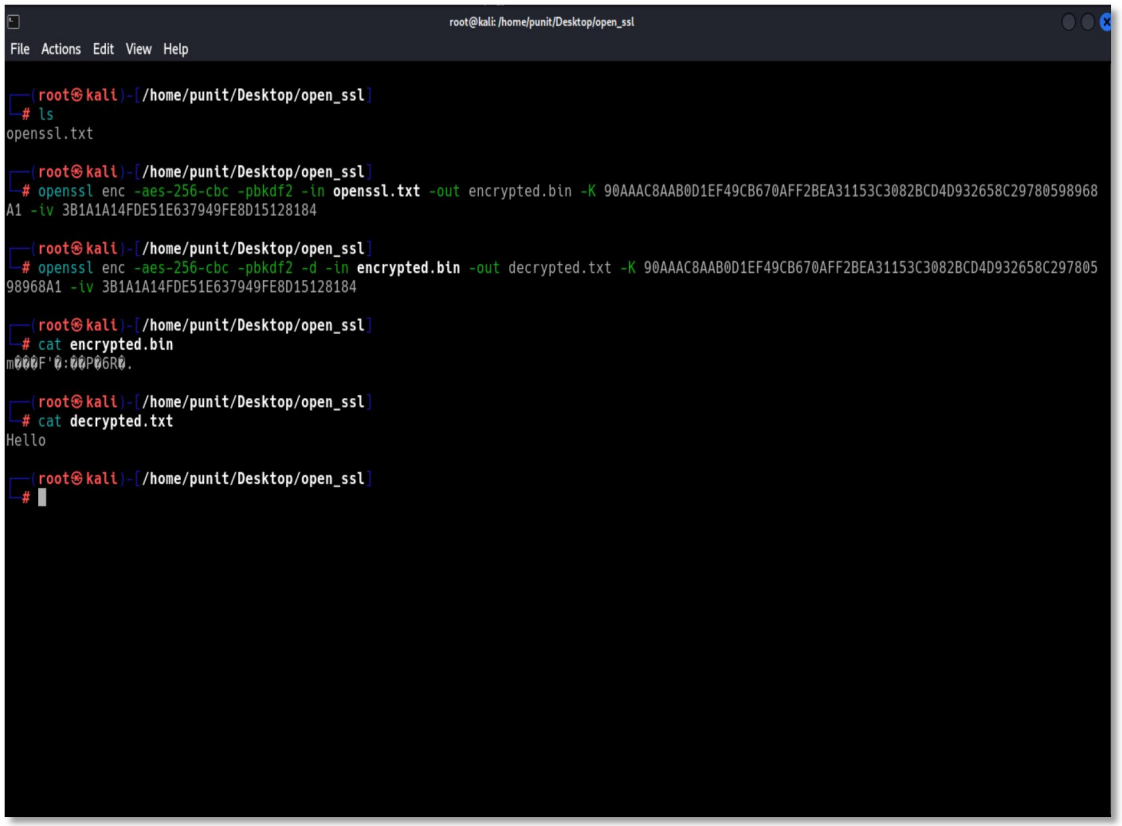
openssl enc -aes-256-cbc -pbkdf2 -in openssl.txt -out encrypted.bin -K <key> -iv <iv>
key=90AAAC8AAB0D1EF49CB670AFF2BEA31153C3082BCD4D932658C29780598968A1
iv =3B1A1A14FDE51E637949FE8D15128184



3. Decrypt the File:

openssl enc -aes-256-cbc -pbkdf2 -d -in encrypted.bin -out decrypted.txt -K 90AAAC8AAB0D1EF49CB670AFF2BEA31153C3082BCD4D932658C29780598968A1 -iv 3B1A1A14FDE51E637949FE8D15128184

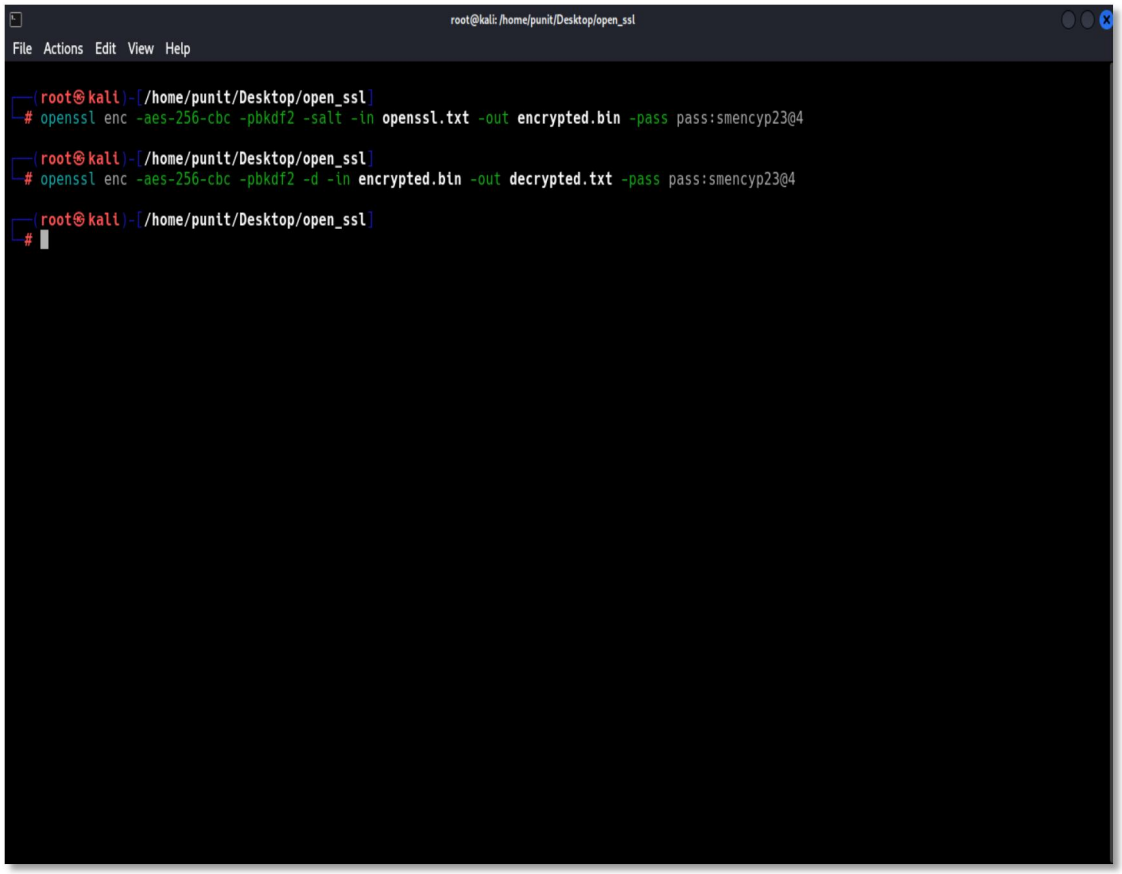




4. Encrypt Using a Password Instead of a Key:

openssl enc -aes-256-cbc -pbkdf2 -salt -in openssl.txt -out encrypted.bin -pass pass:smencyp23@4

openssl enc -aes-256-cbc -pbkdf2 -d -in encrypted.bin -out decrypted.txt -pass pass:smencyp23@4



5. View Encrypted File in Base64(a way to represent binary data as a string of text):

openssl enc -aes-256-cbc -pbkdf2 -base64 -in openssl.txt -out encrypted.b64 -pass pass:smencyp@332

openssl enc -aes-256-cbc -pbkdf2 -d -base64 -in encrypted.b64 -out decrypted.txt -pass pass:smencyp@332