

Security File Sharing System — Task 3 Report

About the Task:

In this hands-on project, I built a secure file sharing system as part of my task. The aim was to simulate a real-world scenario where users can upload and download files securely ; just like in healthcare, legal, or enterprise settings where data confidentiality is crucial.

To achieve this, I used Flask (Python) for the backend and AES encryption to protect files both at rest and during transfer.

Environment Setup

- Operating System: Kali Linux (via VMware)
 - Framework: Python Flask
 - Encryption Tool: PyCryptodome (for AES)
 - IDE/Tools Used: Terminal, Nano Editor, Firefox browser
-

Steps Taken

Set Up Project Folder, Installed flask and pycryptodome

```
Sudo apt install python3-venv
```

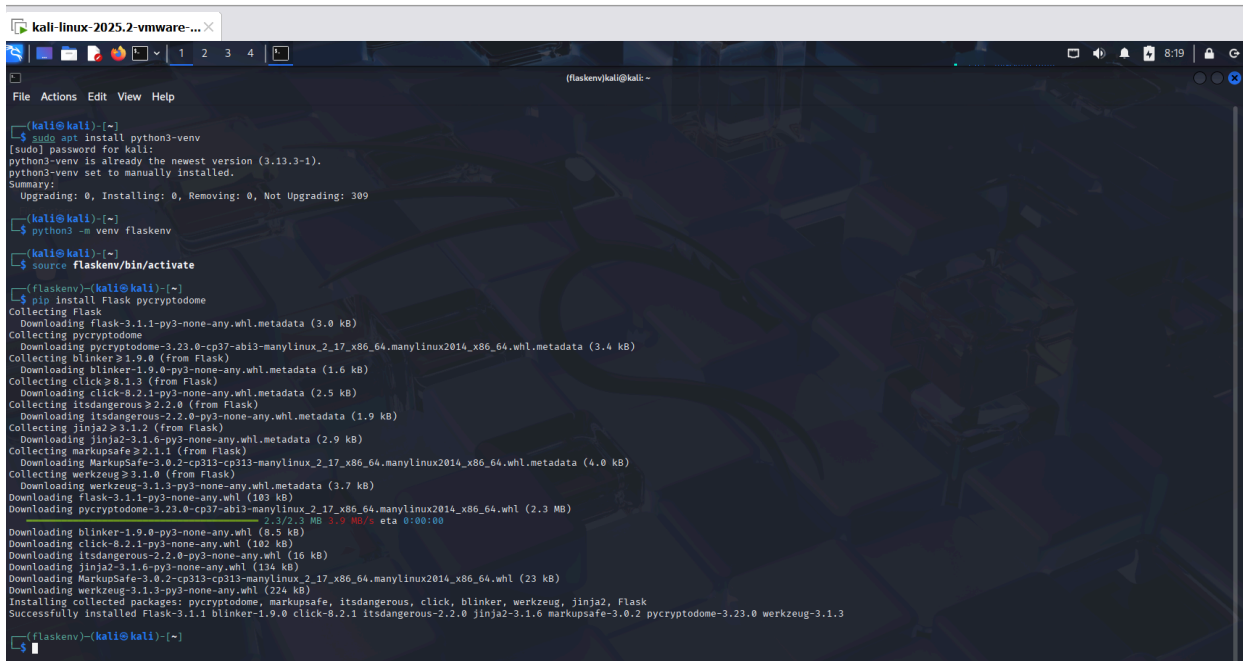
```
source flaskenv/bin/activate
```

```
mkdir secure-file-share
```

```
cd secure-file-share
```

```
python3 -m venv flaskenv
```

```
pip install flask pycryptodome
```



```
kali@kali:~$ sudo apt install python3-venv
[sudo] password for kali:
python3-venv is already the newest version (3.13.3-1).
python3-venv set to manually installed.
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 309

kali@kali:~$ python3 --m venv flaskenv

kali@kali:~$ source flaskenv/bin/activate

(flaskenv)kali@kali:~$ pip install Flask pycryptodome
Collecting Flask
  Downloading flask-3.1.1-py3-none-any.whl.metadata (3.0 kB)
Collecting pycryptodome
  Downloading pycryptodome-3.23.0-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Collecting blinker>=1.9.0 (from Flask)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from Flask)
  Downloading click-8.2.1-py3-none-any.whl.metadata (2.5 kB)
Collecting itsdangerous>=2.2.0 (from Flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from Flask)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from Flask)
  Downloading MarkupSafe-3.0.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.0 kB)
Collecting werkzeug>=3.1.0 (from Flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Downloading flask-3.1.1-py3-none-any.whl (103 kB)
Downloading pycryptodome-3.23.0-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
2.3/2.3 MB 3.9 MB/s eta 0:00:00
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.2.1-py3-none-any.whl (102 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
Downloading MarkupSafe-3.0.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (23 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Installing collected packages: pycryptodome, markupsafe, itsdangerous, click, blinker, werkzeug, jinja2, Flask
Successfully installed Flask-3.1.1 blinker-1.9.0 click-8.2.1 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.2 pycryptodome-3.23.0 werkzeug-3.1.3

(flaskenv)kali@kali:~$
```

Created Flask App Structure

Folders:

- **uploads/**: Where encrypted and decrypted files are stored
- **templates/**: HTML templates folder

Files:

- **app.py**: Main Flask application
- **templates/index.html**: Web interface for file upload/download

```
(flaskenv)kali@kali: ~/secure-file-share
File Actions Edit View Help
Collecting markupsafe>2.1.1 (from flask)
Collecting werkzeug>3.1.0 (from flask)
Collecting flask>3.1.1-py3-none-any.whl (103 kB)
Collecting pycryptodome>3.23.0-cp37-abi3-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (2.3 MB)
Collecting blinker-1.9.0-py3-none-any.whl (6.5 kB)
Collecting click-8.2.1-py3-none-any.whl (102 kB)
Collecting itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting Jinja2-3.1.6-py3-none-any.whl (134 kB)
Collecting MarkupSafe-3.0.2-cp311-cp311-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (23 kB)
Collecting Werkzeug-3.1.3-py3-none-any.whl (224 kB)
Installing collected packages: pycryptodome, markupsafe, itsdangerous, click, blinker, werkzeug, Jinja2, Flask
Successfully installed Flask-3.1.1 blinker-1.9.0 click-8.2.1 itsdangerous-2.2.0 Jinja2-3.1.6 markupsafe-3.0.2 pycryptodome-3.23.0 werkzeug-3.1.3

(flaskenv)kali@kali:~$
$ mkdir secure-file-share
(flaskenv)kali@kali:~$
$ cd secure-file-share
(flaskenv)kali@kali:~/secure-file-share$
$ touch app.py
(flaskenv)kali@kali:~/secure-file-share$
$ mkdir uploads
(flaskenv)kali@kali:~/secure-file-share$
$ touch templates/index.html
touch: cannot touch 'templates/index.html': No such file or directory
(flaskenv)kali@kali:~/secure-file-share$
$ touch templates/index.html
touch: cannot touch 'templates/index.html': No such file or directory
(flaskenv)kali@kali:~/secure-file-share$
$ mkdir templates
(flaskenv)kali@kali:~/secure-file-share$
$ touch templates/index.html
(flaskenv)kali@kali:~/secure-file-share$
$ nano templates/index.html
(flaskenv)kali@kali:~/secure-file-share$
```

```
kali-linux-2025.2 vmware-...
(flaskenv)kali@kali: ~/secure-file-share
File Actions Edit View Help
GNU nano 3.4 templates/index.html
<!DOCTYPE html>
<html>
<head>
<title>Secure File Upload</title>
</head>
<body>
<h2>Upload a File</h2>
<form action="/upload" method="post" enctype="multipart/form-data">
<input type="file" name="file" required>
<button type="submit">Upload</button>
</form>
<h2>Download a File</h2>
<form action="/download" method="post">
<input type="text" name="filename" placeholder="Enter filename" required>
<button type="submit">Download</button>
</form>
</body>
</html>
</html>
```

```
(flaskenv)kali@kali: ~/secure-file-share
File Actions Edit View Help
(flaskenv)kali@kali:~$
$ mkdir secure-file-share
(flaskenv)kali@kali:~$
$ cd secure-file-share
(flaskenv)kali@kali:~/secure-file-share$
$ touch app.py
(flaskenv)kali@kali:~/secure-file-share$
$ mkdir uploads
(flaskenv)kali@kali:~/secure-file-share$
$ mkdir templates
(flaskenv)kali@kali:~/secure-file-share$
$ touch templates/index.html
(flaskenv)kali@kali:~/secure-file-share$
$ nano templates/index.html
[280~nano template/index.html
zsh: bad pattern: [280~nano
(flaskenv)kali@kali:~/secure-file-share$
$ nano templates/index.html
(flaskenv)kali@kali:~/secure-file-share$
$ touch app.py
(flaskenv)kali@kali:~/secure-file-share$
$ nano app.py
```

```
kali-linux-2025.2-vmware-... X
(Flaskenv)kali@kali: ~/secure-file-share

File Actions Edit View Help

GNU nano 3.4 app.py
encrypted = encrypt_file(file_data)
with open(os.path.join(UPLOAD_FOLDER, filename + '.enc'), 'wb') as out:
    out.write(encrypted)
return 'File uploaded and encrypted successfully!'

@app.route('/download', methods=['POST'])
def download():
    filename = request.form['filename']
    enc_path = os.path.join(UPLOAD_FOLDER, filename + '.enc')
    if not os.path.exists(enc_path):
        return 'File not found'
    with open(enc_path, 'rb') as f:
        enc_data = f.read()
    decrypted = decrypt_file(enc_data)
    dec_path = os.path.join(UPLOAD_FOLDER, filename)
    with open(dec_path, 'wb') as f:
        f.write(decrypted)
    return send_file(dec_path, as_attachment=True)

if __name__ == '__main__':
    app.run(debug=True)
```

```
(Flaskenv)kali@kali:~/secure-file-share$ python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 347-275-278
127.0.0.1 - - [11/Jul/2025 17:23:54] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Jul/2025 17:25:11] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [11/Jul/2025 17:26:18] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Jul/2025 17:28:08] "POST /download HTTP/1.1" 200 -
^C

(Flaskenv)kali@kali:~/secure-file-share$ ls uploads
'SOC_Task2_Sample_Logs(1).csv.enc'
```

Built Secure Upload Logic

- Used `request.files` to grab the file
- Encrypted file using **AES (EAX Mode)**
- Stored `.enc` files in an `uploads/` folder

Implemented Secure Download Logic

- User enters the original filename
- The app finds the encrypted `.enc` version
- Decrypts it and returns the original file

Created User Interface

- A simple HTML form to:
 - Upload files
 - Download files

Tested the System

- Uploaded `.csv`, `.txt`, and other files
- Verified successful encryption and decryption
- Monitored file changes in the `uploads/` directory

AES Encryption Overview

- Used `PyCryptodome`'s AES in **EAX mode**
- Key was static for demo: `b'ThisIsASecretKey'`

Encrypted file stored as:

`nonce + tag + ciphertext`

- During download:

- Nonce and tag verified
- Decryption returns the original file

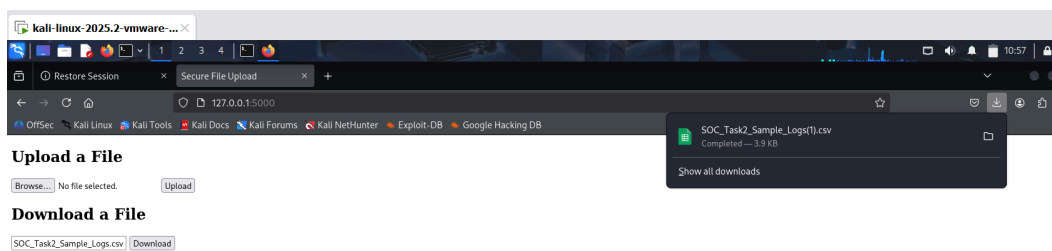
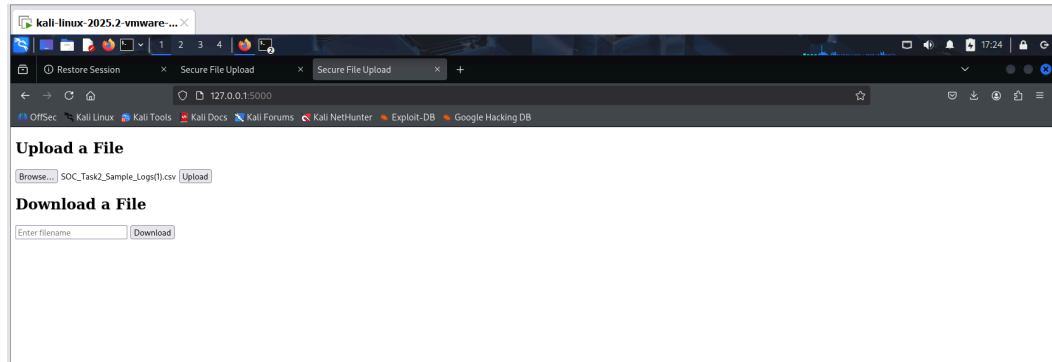
Project Structure

```
secure-file-share/  
├─ app.py  
├─ uploads/  
|   └─ filename.enc  
├─ templates/  
|   └─ index.html
```

Visual Evidence

- Upload form success message
- Encrypted file inside `uploads/`
- Download form decrypted successfully
- Flask running at `127.0.0.1:5000`

```
(flaskenv)-(kali@kali)-[~/secure-file-share]  
└─$ python3 app.py  
* Serving Flask app 'app'  
* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
* Restarting with stat  
* Debugger is active!  
* Debugger PIN: 347-275-278  
127.0.0.1 - - [11/Jul/2025 17:23:54] "GET / HTTP/1.1" 200 -  
127.0.0.1 - - [11/Jul/2025 17:25:11] "POST /upload HTTP/1.1" 200 -  
127.0.0.1 - - [11/Jul/2025 17:26:18] "GET / HTTP/1.1" 200 -  
127.0.0.1 - - [11/Jul/2025 17:28:08] "POST /download HTTP/1.1" 200 -  
^C  
  
(flaskenv)-(kali@kali)-[~/secure-file-share]  
└─$ ls uploads  
SOC_Task2_Sample_Logs(1).csv.enc  
  
(flaskenv)-(kali@kali)-[~/secure-file-share]
```



Key Takeaways

- Learned Flask backend and routing
- Understood AES symmetric encryption workflow
- Built a fully working secure file transfer system
- Practiced file I/O, data security, and Python logic
- Improved confidence in secure app development

