PGP Encryption using Python

Sushank Saini
July 9, 2024

GitHub Link:

Overview

This Python script encrypts and decrypts a file using **PGP encryption-decryption** implemented through **GNU Privacy Guard wrapper** for Python. Once the file is encrypted, it is uploaded to an FTP server. Then, the file is decrypted after being downloaded from the FTP server. The FTP protocol is implemented in the script through **ftplib module**. Finally, the script uploads the file and its decrypted contents to a **MySQL database**.

Best Practices Used

The following best practices have been used to create a **flexible and maintainable code**.

1. *Single Responsibility Principle*: Each method of this script does one and only one thing to have **less coupling and high cohesion**.
2. *Logging*: Logging has been implemented using the **logging module (**see **Figure 1).** This would help with easy troubleshooting of the issues in the code.
3. *Usage of Environment Variables*: To avoid hardcoding of variables directly into the code **.env file** has been used to store the variables and their values (see **Figure 2**).
4. *Usage of Stored Procedures*: All the SQL code has been executed using the stored procedure instead of hard coding the SQL statements in the code itself (see **Figure 3**).

Figure 1: Implementation of Logging

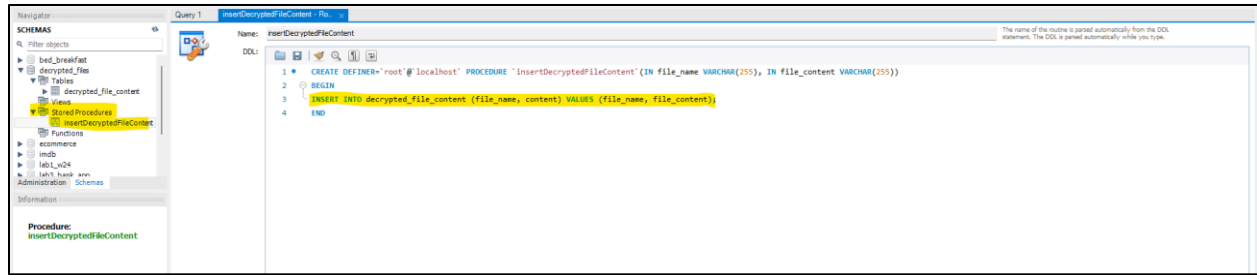*Figure 2: Snapshot of .env file*

Figure 3: Stored Procedure

Program Execution

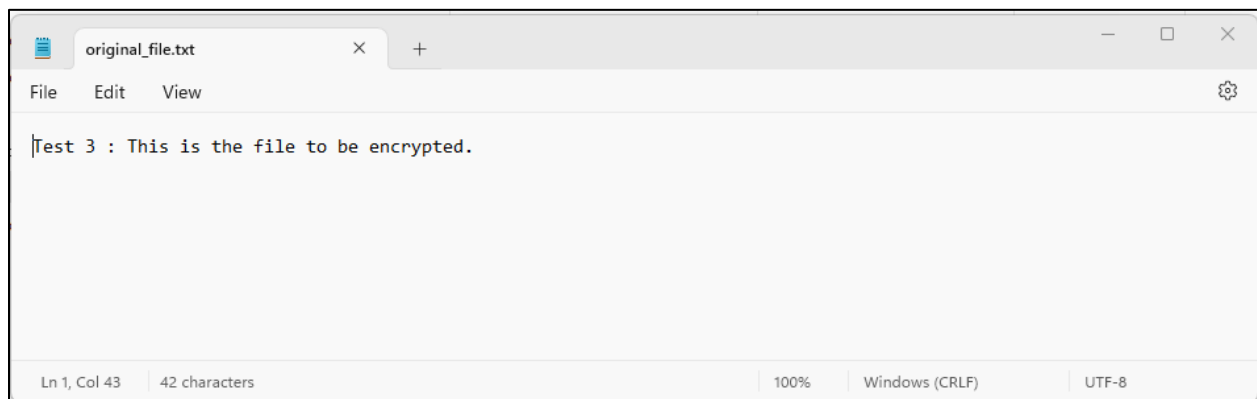The script first reads the original file (**Figure 4**) that is to be encrypted.

Figure 4: Original text file

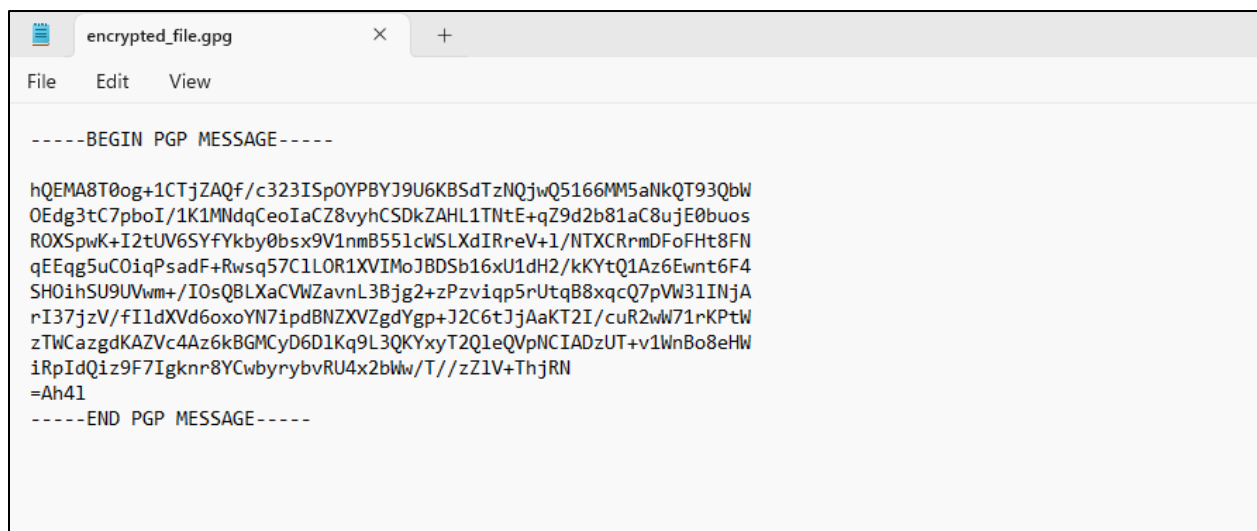After this the file is encrypted (**Figure 5**) and uploaded to the FTP server (**Figure 6**).

Figure 5: Encrypted File

Figure 6:File Upload on FTP

Then the file is downloaded from the FTP server so that it can be decrypted. Once decrypted, the file's content is pushed to a MySQL database (**Figure 7**).
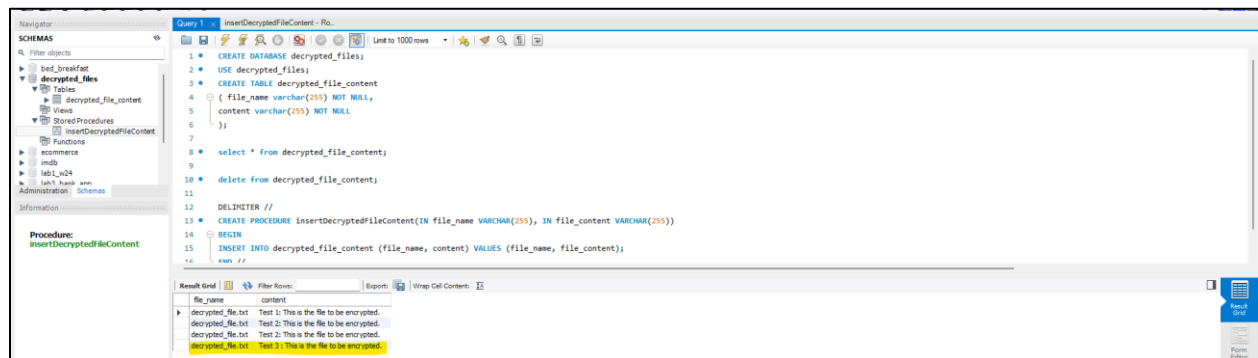

Figure 7: File Upload to Database