Assignment 2 Part 3

Three significant vulnerabilities in my original reference monitor were:

1. **Lack of Multithreading Safety**: My original code did not consider multithreading issues. This lack of coordination led to conflicts. When multiple threads access the class concurrently, it leads to race conditions. This causes data corruption and unpredictable behavior.

2. **Offset Validation**: My original code did not adequately check if the offset parameter in the writeat and undo functions was valid. This led to offsets going beyond the End of File (EOF) and negative offsets, causing exceptions or unexpected results.

3. **Exceptions:** My original code did not raise the exceptions as it should have. For example, RepyArgumentError for negative offset and FileClosedError when the file is already closed.

To fix these issues, I added safety measures to handle multiple actions, checks to ensure that actions are within the file's limits, and exceptions. This makes the code more reliable and easier to work with. Therefore, to address the vulnerabilities mentioned above, I made the following changes to the reference monitor:

Multithreading: I updated the writeat and close functions to include a locking mechanism using a createlock object. This ensures that only one thread can perform operations on the file at a time. This modification prevents race conditions and maintains data integrity during concurrent access. Introducing locks confirms thread safety, preventing data corruption and enhancing reliability.

Offset Validation: In the revised code, I updated both writeat and undo functions to check if the provided offset is non-negative, as well as a check for offset going beyond the limits of the file. If the offset is less than zero, a RepyArgumentError is raised, ensuring the offset is within valid bounds. The size of the data in the file is also checked before a write operation to ensure that the offsets do not exceed the file size.

Exceptions: I added valid exceptions to the code to raise appropriate exceptions whenever it encountered abnormal behavior.

In conclusion, the vulnerabilities in the reference monitor were addressed by adding a locking mechanism to provide multithreading safety and by validating offsets to prevent them from exceeding the EOF. Exception handling was also done to recreate proper repy behavior. These changes enhanced the security and dependability of code and improved protection against data corruption and exceptions, making it a more secure and reliable component in the security layer.