Nikhita Dhenuvakonda, nd2661

Write up on vulnerabilities in original reference monitor and how I fixed them

The vulnerabilities in the initial code and the fixes implemented in the revised version are primarily related to error handling, data validation, and the undo operation.

Code Enhancements:

1. Implementation of Locks:
   To address potential race conditions and ensure thread safety, locks have been implemented in critical sections of the code. By utilizing locks, concurrent access to shared resources is properly controlled, minimizing the risk of data corruption and inconsistent program states.

2. Offset Error Handling:
   An offset error handling mechanism has been introduced to validate the offset value during file read and write operations. The code now checks if the offset is within the acceptable range and raises appropriate exceptions, such as RepyArgumentError and SeekPastEndOfFileError, when necessary. This ensures that file operations are performed securely and accurately.

3. Exception Handling Alignment:
   In order to align the code's behavior with the underlying Repy API, adjustments have been made to the exceptions raised in exceptional scenarios. This change ensures that the security layer consistently reflects the expected behavior of the Repy API, providing accurate feedback and enabling proper error handling.

4. File Size Check:
   To mitigate potential memory issues and maintain system stability, a file size check mechanism has been implemented. This enhancement verifies whether the size of the file being accessed or modified is within acceptable limits. If the file size exceeds the specified threshold, appropriate actions are taken, such as truncating or rejecting the operation, to prevent resource exhaustion and potential security vulnerabilities.

5. FileClosedError Implementation:
   To improve error handling and provide informative feedback, a FileClosedError has been implemented. This error is thrown when an operation(read/write/undo) is attempted on a closed file. By detecting and handling this specific error scenario, the code can gracefully handle closed file situations, preventing potential data corruption or unauthorized access attempts.

6. Validation for Negative Bytes:
   To ensure the integrity of file operations, a validation check has been added to the readat method. If the bytes parameter is less than zero, a RepyArgumentError exception is raised.

Conclusion:
The enhancements made to reference monitor significantly strengthen its undo functionality and overall security. By implementing locks, handling offset errors, aligning exception handling, introducing file size checks, and implementing FileClosedError, the code demonstrates improved resilience against potential attacks and misuse. These enhancements ensure that the security layer operates securely and reliably, protecting sensitive data and maintaining system integrity.