Name: Kushal Kothari
Kk5268

Vulnerabilities in the Initial Code

Insufficient Handling of Undo Functionality: The initial code did not effectively handle the undo functionality, allowing potential attackers to manipulate the system's state easily.

Lack of Offset Validation: Offset values in the writeat() function were not adequately validated, leading to scenarios where negative offsets or writing past the End of File (EOF) were possible.

Inaccurate File Length Calculation: There was an issue with calculating the length of an existing file, which could lead to incorrect handling of offsets.

Exception Precedence: Exceptions were not thrown with the correct precedence, potentially leading to unexpected behavior when multiple exceptions could apply.

Race Condition Vulnerability: The code did not address race conditions when multiple processes accessed the same function block simultaneously.

Enhancements and Fixes

Improved Undo Functionality: In the updated code, we introduced an enhanced undo_write() function to handle the undo functionality more effectively. It allows reverting pending writes and ensures data consistency.

Offset Validation: The new code includes offset validation in the write_at() function to prevent negative offsets and writing past EOF. It raises appropriate exceptions (e.g., RepyArgumentError and SeekPastEndOfFileError) to handle offset-related issues.

Accurate File Length Calculation: We addressed the file length calculation issue by maintaining two variables: pending_eof and previous_eof, which help accurately track the file's length and handle offsets.

Exception Precedence Management: The code now correctly manages exception precedence, ensuring that exceptions are thrown in the correct order, preventing unexpected behavior.

Race Condition Mitigation: To mitigate race conditions, we introduced a global lock, ensuring that only one process can access the critical function block at a time.