Gayatri Tyagi

11/2/23

Assignment 2.3 Report

The reference monitor which I submitted initially covered some basic undo functionality, however I was missing end-of-file tracking and my methods did not throw the expected exceptions in certain conditions.

I added conditions to my writeat() and close() methods so that they would throw the appropriate exceptions, aligned with the behavior of standard repy writeat() and close() commands. I ordered the conditions so that exceptions would be thrown in the correct order of precedence based on the documentation. I also added end-of-file tracking so that subsequent writes could not be done past the end of the file, even if the write is technically still pending. This involved adding parameters like file_length and bytes_written. While file_length stores the amount of data completely written to the file, bytes_written stores the number of bytes the user has asked to write so far. The difference here is that file_length does not account for any changes due to pending writes while bytes_written does. This allows for updating bytes_written every time a writeat() command is called, updating file_length only when data is committed to the file, and changing EOF appropriately on an undo(). An alternative to this kind of tracking of file length would be to read the whole file on every undo, but I was concerned that that would be less efficient. Lastly, in my readat() function, I fixed an issue where in some cases, a variable would be returned that hadn't been defined.

Overall, I learned a lot during my work on the assignment in terms of considering race conditions and appropriate error handling when modifying and adding functions. Seeing other people's attacks and reference monitors was a great opportunity to learn and expand my idea of what attacks and vulnerabilities might look like.