CS-UY 3923: A2-3

Syntax: I started off by correcting my syntax error. Nothing would run so any attack would've passed through it, which made my security layer break everything from the get-go. Instead of using the keyword 'with' to handle my locks, I acquired a lock at the beginning of every method, and released them as I went. This meant that anytime I raised an exception, I would also release the lock before the exception as to avoid a gridlock.

EOF Size: As well, my EOF tracking was a little off after an undo – it would return the EOF size before the undo, but before the valid write as well. This managed to allow invalid writeat()'s through my security layer which made it vulnerable to EOF attacks. So, the way I kept track of the EOF size, was that if the current length of the writing data + the offset was greater than the previous EOF size, then it's updated. I also kept track of the file_size before committing and reset the actual file_size equal to the old_file_size after an undo to avoid the issue I addressed before.

Error Hierarchy: Furthermore, I found out that my exception hierarchy was wrong. This made my program misleading as to what kind of error should've been addressed, making it vulnerable to attacks. Instead of raising the RepyArgumentError first, I raised FileClosed. I rearranged my exceptions in writeat() to deal with this. (New Order: RepyArgumentError, FileClosedError, EOF)

Locks & Threading Problems: Majority of the other attackcases that were successful against my monitor related to threading. I added locks in the appropriate places like writeat() and undo(). We didn't care about readat() taking turns since we weren't modifying the file so I didn't add locks there.

Close: close() was another scenario that I had to fix with locks. So before, I had no locks on it because I was getting into a deadlock when I added a lock and tried to commit the writeat(). So instead, I added the locks and this time committed the writeat() using the original writeat() instead of my wrapper method. Then raised any exceptions with a try/finally block to release the lock afterwards.