The biggest challenge of deploying the security layer for me is understanding the security policies. My security layers' vulnerabilities mainly come from misunderstanding the policies.

My biggest security issue is the exception catching on writeat() function. Since the real writeat() action happens in the next call on writeat() or close(), the exceptions of writeat() will not appear immediately and will disappear after undo(). Formerly, I think this is OK since exceptions about writeat() will be raised in next call or close() anyway and if the action is undoed no need to raise exceptions. However, from the test cases I realize the writeat exception should be raised in real time. Thus, I add three exception catchers before the code in writeat().

I catched RepyArgumentError by checking on data and offset. I catched FileClosedError by checking on the closed status of the file: implementing by assigning a variable closed to False every time opening the file and setting it to True just before close() action. I catched SeekPastEndOfFileError by keeping track of the ending index of the file: read the file and get the end index every time open the file; compare the offset and ending index everytime call the writeat(); update ending index at the end of writeat() and undo().

My second security issue is on multithread. In my original security layer, I apply the locks to make sure writeat(), readat(), undo(), and close() execute atomically. However, in one off the test case, it suggest that a writeat() call in a thread can commit the real writeat() action even if the pending data and offset come from another thread, but the undo() call in a thread can only undo its own writeat() call in the same thread. This part is not really mentioned in the policy, but I modified it anyway. I implement a variable to record the thread name of pending data and offset. Undo() can only process if its thread name is the same as pending data and offset.

In short, security is all about corner cases. It is true that some cases I never thought about, and these test cases can really help the improvement of security layers.