A lot of the vulnerabilities that happened within my reference monitor were because I either didn't raise a lot of the expected exceptions/errors based on the test cases provided by my classmates, I didn't input locks into my reference monitors, and I didn't properly handle writeats in terms of dealing with the given offsets. In order to fix the expected exceptions/errors, I had to read through the API documentation one more time, and within both my readats and writeats I included if-else checks so that when the appropriate situation came up, the correct error was raised. I also had to check and make sure that the appropriate error was checked for first in order to mimic the actual writeat functionality. This resolved my issues regarding the errors being raised. The next issue came up with the locks, I didn't implement any at all during the first assignment, so this time I created my lock in the init function and then made sure to acquire and release the locks within the readat, writeat, undo, and close functions in the appropriate locations. This way when threads are running at the same time, the first thread conducting the particular function won't be interrupted by another thread happening at the same time and cause errors to pop up. When these conflicts don't occur the threads run as expected thanks to the locks and we get the output we want based on the test cases. Regarding handling cases of multiple writes with different offset scenarios there was a lot I had to do in order to fix the cases. I first log out different variables within my reference monitor to see what scenarios are causing the error and why. I would then run through my code logic step by step on paper to see which part of the logic went wrong which was really helpful. I then changed and added if statements within my writeat and ensured that when checking what the new offset would be, I was updating it properly so that when I checked with the offset given in the test case it wasn't an End of File situation. By solving these three major issues, I was able to fix the attack cases that did penetrate my attack case.