Esther Wang

The vulnerabilities that went through the reference monitors had to do with not setting locks when running the attack cases. I fixed this issue by creating a lock feature inside my init function and acquiring the lock whenever a writeat(), readat(), and undo() are done. In this way, I am able to make sure that other threads can not use the function until the current thread is done running. I also created a buffer so that the writeat() had some time to check if they were able to complete the command. I put all my code in a try statement so that exceptions could be caught if there were any errors with the attack cases. Finally, I released the lock once an action was complete to signify if the thread was done. Another mistake in my reference monitor was not checking the offsets right. I did try to see if my offset was overwriting the previous offset or if the new offset went over the original offset. I checked to see if the offset was enough to hold the information in the writeat(). I also checked if there was a negative offset. I raised the appropriate errors for each. I raised errors like SeekPastEndOfFileError() and RepyArgumentError() for checking offsets. I also raised FileClosedError() error to see if the file is closed. If the file was closed then I would restrict the writeat() and undo(), but still allow the readat(). After each writeat(), readat(), and undo() I checked to make sure that all the needed variables that need to be set to None are set to None. I also created a fileSize variable to check the size of the file after every action so that I can keep track of the action would be valid for the file size of each file.

There were certain cases that I didn't check before each action and I implemented a check for all of those cases with if statements. I checked if the reading bytes were valid numbers, if the offset was less than zero, see if LPFile was empty using None, if the file was closed during a writeat(), and if the writeat() was performing a valid writeat() of string values. All those cases were checked to see if they needed to raise an error. The errors were raised in the above format.