Name: Adittya Mittal **Net ID:** am14079

CS-GY 6813: Information Security and Privacy

Assignment: 2 – Part: 3

Based on testing my reference monitor using all the attacks submitted by other students, I was able to identify a few vulnerabilities in my reference monitor. These vulnerabilities ranged from a basic negative offset check to a complex multithreading lock failure and are described as follows:

1) **Negative Write Offset:**
   I realized that my reference monitor was not checking for a valid offset value on the writeat() where it doesn't yet write to the file and stores the data to be written as pending data. To fix this, I added a check for valid offset value and raised an exception if it failed.
2) **SeekPastEndOfFileError:**
   My reference monitor was not keeping track of the updated offset value on the writeat() which stores data to be written as pending data. I fixed this by adding a variable to keep track of the updated file size when data is added to the pending variables. I also reset this new file size value on undo() function calls.
3) **FileClosedError:**
   I realized that the undo() function was failing when the file was closed, so to counter this, I added a flag variable which gets set to "true" whenever the file gets closed and is initialized to "false" when the file object is first instantiated. Then I simply check for the flag value before carrying out my undo operations
4) **Multithreading:**
   I saw that my reference monitor was not working as expected for a few attack programs which used multithreading. To fix this, I introduced locks on all functions and made sure to release them in the "finally" block so as to not cause a deadlock if the code in the "try" block raises an exception.
5) **Order of Raised Exception:**
   I fixed the order of exceptions being thrown to be in accordance with the order of exceptions thrown by the RepyV2 API.

Apart from these, there were a few other changes I made to make my reference monitor more robust, such as checking for valid input type for the offset (integer) and data (string) and checking for data and offset to be present before setting the value.

Other than the reported invalid attacks, my reference monitor is now able to withstand all other attacks provided by all the students!