Dinh Khanh Duong
Computer Security CS3923 / 6813

Assignment 2.3 Reflection

For this assignment, a lot of the mistakes were realized when I was working on 2.2 as I saw how different students implemented the code. My understanding of secure file operations has deepened that lead to revisions in how I implemented my code. I especially came to understand the importance of edge cases and how a layer operates in certain contexts. I was amazed at some ways other students implemented their code with protections I didn't think about or sometimes even knew was possible.

My biggest mistake was getting confused close to the deadline and thinking the code had to have multiple undos even though it didn't. I also realized that I did not implement any protections against thread attacks as it didn't even occur to me. Those were the first fixes I wanted to take care of right away. Below are the changes explained in more detail that should encompass all the changes that I made. Hopefully I didn't forget to mention any.

The first change is that I modified and expanded the checks for input for the write and read functions, specifically making sure they are integers and not negative numbers. This is important as proper validation helps prevent the attacks that exploit parameter abuse. This just makes sure that what we get is what is expected and is one of the basic or fundamental practices.

Another change is tracking the end of the file with the max size variable as I realized that the old code didn't handle cases that attempted to read or write past the end of the file. To combat this additional checks were added to keep track of and check that those cases don't happen. This should help with attacks that aim to exploit overflow vulnerabilities and helps with integrity and maybe even unwanted memory access. In the third iteration I removed this variable for a simpler and direct approach to managing file boundaries with the variable file_size as the previous change didn't consider the case of a write operation extending beyond the current file size.

One of the bigger challenges was trying to combat threading by introducing locks in the operations to make sure there is thread safety. The locks should prevent race conditions and help protect the integrity of the data and its consistency too.

Undo was changed from a stack to simply keeping track of the latest write and changing when things are committed to file instead. Lastly, the close method has been augmented to perform an automatic undo of the last write operation if the file is closed without an explicit undo. This acts as a fail-safe to ensure that the last state of the file remains consistent with user expectations.