Aurora Cruci: 2.3 Writeup

Initial Code Vulnerabilities

1. I didn't have any form of thread-safety, so if multiple threads tried to access and modify a text file simultaneously, nothing would have stopped it from forcing my code to act in an undefined manner.
2. There was nothing checking to make sure an attacker couldn't double-close a file. I had checks to make sure they couldn't double-open a file, but for some reason I had nothing in place to prevent undefined behavior for closing.
3. My code didn't check for negative offset values or invalid byte sizes in the "readat" and "writeat" methods, so any undefined behavior that could have resulted from that was left unaddressed.

How I Improved The Code

1. I added locks to ensure that only one thread can access and modify my code at a time. The lock is created at the initialization; it is acquired at the start of each method and subsequently released when the method finishes running.
2. The "is_closed" flag was used to function for checking if the file was already closed before attempting to close again, similarly to how it was used in opening the file. The "close" method checks the flag before continuing, such that it simply returns if the file is already closed.
3. The "readat" and "writeat" methods were updated to include checks for negative offset values and invalid byte sizes, such that it simply returns if any of the aforementioned cases is true.