Ricky Zapata
Professor Cappos
Computer Security

Improvements to Security Layer

One of my security layer's most lacking areas was its protection against multi-thread attacks. In order to defend against multi-thread attacks and improve my security I implemented the use of createlock(), acquire(), and release(). By creating a lock in my constructor using createlock() as well as making sure to acquire the lock once I need it (using acquire(True)) and release the lock at the end of each process (using release()), I was able to ensure the safety of the thread execution and defend against multi-thread attacks where multiple threads of execution may cause errors in a security layer. In conjunction with this, I also used try: and finally: blocks in order to not only maintain the efficacy of the locks that I implemented as well as improving my error handling. With my improved security layer, I only allow valid operations to occur and invalid operations fail silently. I implemented error handling in this way to make sure that attackers would not be able to attempt attacks and use statements printed by the security layer through logs or exceptions raised in order to complexify or improve their attack strategy.

Another big issue in my previous security layer was my lack of error handling in the close() method. When close() was called previously, my initial security layer would not do any error handling and would close. This would not handle the case of if the file has already been closed. With my new security layer, I not only use locks but I also make sure to check whether the file has already been closed and if it has then the call to close() fails silently. On the topic of error handling, I decided to separate the error cases that I previously grouped together. This is useful in the event that one wants to come back to the security layer to improve it more, implementing a different security strategy as opposed to allowing errors to fail silently.