Zaki Zahid Maqbool (N13108415)

31st October 2023

Report (Fixing the Security Layer)

There were several vulnerabilities in the reference monitor which I turned in initially and I tried to fix and make improvements in the code. First, while fixing the security layer, there is input validation for 'filename' and 'create' and ensured that 'filename' is a string and 'create' is a Boolean. This clarifies that the input parameters are of the correct types.

The patched reference monitor includes proper error handling. Initially there was no error handling for scenarios like writing with a negative offset, writing beyond the end of the file, etc. Such error checks are essential to prevent data corruption and unexpected behavior. It uses *try* and *except* blocks to catch exceptions that may occur during file operations for example opening, reading, writing, or closing a file. If an exception occurs, it will raise an informative error message which would make it easier to diagnose and fix the issue.

It includes an updated "writeat' method which now includes proper error handling for writing to the file. It checks if 'self.pending_data' and self.pending_offset' are not None before writing. And if an exception occurs during writing, it is properly handled.

Similarly, the 'close' method is also updated and includes proper error handling for writing pending data before closing the file. It checks if 'self.pending_data' and 'self.pending_offset' are not None before writing. And if an exception occurs during writing or closing, it will raise an informative message.

There are also some general improvements made in the patched reference monitor such as attribute assignment. In the new reference monitor, I use the is not None condition instead of != None for attribute assignments such as self.pending_data is not None. This is a more pythonic way of checking for None.
Other than that, the patched code itself contains comments which explains the purpose of it and these comments help improve code readability.
Last but not the least is for clarity, the code includes additional comments on the purpose of each section and error handling.