Name: Abhay Garg
Net ID: N19973700

ISP Assignment - 2.3

Vulnerabilities present in my code:

Case-1: My reference monitor was not handling Exceptions scenarios. For example, if the attacker is trying to write at a negative offset, the reference monitor wasn't handling this case. This was making the reference monitor not resilient.

Case-2: When the attacker tries to write on a closed file, the reference monitor was not identifying such a scenario.

Case-3: When the attacker tries to write at the offset greater than the file size, the reference monitor was not handling this attack.

Case-4: The reference monitor was not updating the file size post write/undo.

Potential fix for the vulnerabilities:

1. For the Case-1, after adding an if condition to check whether the offset is negative. If yes, then raise the RepyArgumentError exception.

2. For the Case-2, after checking the negative offset condition, the reference monitor checks if the attacker/user is writing the file without opening it. If so, it raises an exception "FileClosedError" to let the attacker/user know that this operation is not valid.

3. For the case-3, the reference monitor compares the offset and the file size. If offset is greater than the file size, it raises an exception "SeekPastEndOfFileError" to let attacker/user know that it's an invalid operation.

4. For the Case-4, the attacker performs write and undo operations. For example, the attacker performs a write operation of 10 bytes in an empty file and then does an undo operation, now ideally file size should be changed to zero after undo operation. Before the fix, the attacker was able to perform the write operation at offset 10, which is not a valid scenario. Now, after the fix, the file size changes to zero or to the last file size.

After fixing these 4 vulnerabilities, my reference monitor became more resilient and has become immune to all attacks. In my reference monitor, the locking ensures that the thread and multi read-write operations are getting executed without any issues.