Reference Monitor Reflection

I made a lot of mistakes and omissions when coding my original reference monitor. My efforts were concentrated on the writeat and undo functions and how they would impacted. I devoted far less attention to functions like readat, open and close. In addition to the basic functionality, my program handles threaded attacks well due to my use of locks on every one of my functions. My most glaring mistake was neglecting checking file EOF. I had just assumed that when reading and writing, the underlying API would handle EOF properly. I did not take into consideration how having the undo function would interfere with EOF and that I would have to keep track of my own EOF marker. This led to my program failing every EOF attack provided. Furthermore, I also didn't handle exceptions properly for all of my functions. For some of my functions, I would not release the lock after throwing an exception. Therefore, if someone's attack cases managed to catch the exception and then continued execution. My program would fail. This was a lot less common than the EOF attacks. To fix the issues, I added an EOF tracker and changed all my functions to use try-except-finally statements to properly release the locks. I also hardened readat, open and close, because I was neglecting them before.

I spent some time while writing my attack cases to try and escape the repy pyjail. Do there exist any known ways to escape repy?