Michael Young (myoung10)

# Engagement 8-2

A race condition is a situation in programming where 2+ processes are running concurrently and there is an arbitrary completion for each task. I’ve actually ran into this problem quite a bit, and they are extraordinarily difficult to detect because they cannot be consistently replicated. The situation I had at my last job was in counting how many files had been transferred. When a client uploaded, i.e. 10,000 files, multiple threads would spin up to upload as well as multiple back-end processes to process the meta data on each upload. After the file had been processed, a counter was incremented. In order to increment the counter, the program would first have to understand the current count. In this case, if 2 files completed at the same time, they would both increment the same value, i.e. process 1 finishes, checks the current count = 0, process 2 finishes at the same time, checks the current count = 0, and both increment to 1, even though 2 files were actually processed. This is generally fixed with mutx locks which say, “I am reading this value, no one else can read it/write to it, and when I’m done, I’ll release this lock so other processes can read/write to it”. It essentially adds a degree of sequential-ness to an otherwise parallel/concurrent process.

An attacker would not be able to really derive a lot of use out of the race condition above—aside from aggravating clients. A silly example might be one where a concurrent process is set-up for authentication and authorization. These should be sequential processes, as you should not receive authorization permissions until you are authenticated. An attacker could exploit this by either making the authentication finish slower than the authorization request, or by jamming the submit button several times and hoping the race condition switches between authentication/authorization randomly. Maybe a more realistic example may be if I were to send 2 bank transfers at once. To my point earlier about mutex lock, if I were to send 2 bank transfers at the exact same time for $100, and my account is $1,000, and each process saw my balance at $1,000 without a mutex lock, my balance would become $900 instead of $800 for the 2 $100 transfers. This would (should) not impact the recipients.

I think it is very likely an attacker could leverage this issue, sometimes by accident. The reason why I suggest this is so likely is that they are not immediately known as a problem and are difficult bugs for a developer to diagnose.