Assignment 3

He Fu(SID: 012543440)
Rui Guo(SID: 012478973)

1. What happens if the file being requested is not inside the chroot? Does it matter?
Answer:

It doesn't matter if the file being requested is not inside the chroot for this assignment. Because we start the program with root access and had opened the file before fork/exec, which means the program still has root privilege and can open the requested file.

2. Explain the mechanism you are using to pass file descriptors between parent and child processes.
Answer:

In this assignment, both the listening port descriptor and file descriptor need to be passed from the parent process to the child process, but after the fork function, the child process still has access to the parameters. To pass the file descriptors between parent and child processes, we use a "," to separate these two file descriptors in the command line and pass them into the program. If the command line contains ",", that means both the listening port descriptor and file descriptor are passed in, so we need to parse them into two different parts and process them. Otherwise, if there is no "," in the command line, that means there is only the listening port descriptor passed in.

3. What happens if the file size of the disk file exceeds the size of the client's hardcoded buffer? Does the client have sufficient checks to ensure no buffer overruns occur?
Answer:

In this assignment, the client only read 1024 bits (the hardcoded buffer size), so if the file size of the disk file exceeds the size of the client's hardcoded buffer size, it would try to read the first 1024 bits data and ignore the rest part.
The client checks up to 1024 bits of data. However, we don't think it is sufficient because it is not a safe way to read data. The better way is to get the size of input data and then dynamically allocate memory to ensure the data is fully read and buffer overflow is avoided.