The vulnerability in the server program comes from the line

read(fd, &(s.nbytes), sizeof(s));. This line reads sizeof(s) bytes into the buffer starting at &(s.nbytes) from the file descriptor fd. The intention of this line is to only read nbytes into the buffer starting at &(s.nbytes), but nbytes+4 are read in, since the struct s also contains a function pointer which takes up 4 bytes. This allows more pages to be received than intended, making the buffer vulnerable to overflowing. This line can be rewritten as read(fd, &(s.nbytes), sizeof(s.nbytes)); in order to prevent this from occurring. This limits the amount of bytes written in to the buffer starting at &(s.nbytes) to s.nbytes. Now when attempting to run the exploit, the attacker can no longer change s.fp and therefore cannot execute the malicious payload. When the exploited client program is run, the client simply prints “okay” and a newline, and the server continues on without giving access to the attacker.