Hand Rolled Cryptex

Useful C Functions

1. int open(const char *pathname, int flags)
2. ssize_t read(int fd, void buf[.count], size_t count)
3. ssize_t write(int fd, const void buf[.count], size_t count)

Explanation

I completed this challenge using mainly static analysis. This program runs a function for every question. For the first 2 questions, the functions that run should return an int >=0 for the user to advance.

First question

The program takes two inputs for the first question: a string (let's call it str_input) and an integer (lets call it num). It then opens (see 1.) a file with the name str_input and the flag set to num. As long as the user gives a valid name of a file on the server (the flag can be set to 0, for read only), the user will pass this part of the challenge.

Second question

For the second question, the program takes in an input, performs a bitwise NOT on it, and XORs it with 0b11001001. The result of these operations is used as the file descriptor in the read function (see 2.). As long as the user gives an input such that it is a valid file descriptor after being manipulated, the user will pass this part of the challenge.

Third question

For the last question, the function reads in an input, casts it to an int, and returns the int or a -1. The return value of this function is used as the file descriptor for the ending write function (see 3.). If an input of 2 is given using pwntools (where the input can be sent in as bytes), then the contents of the buffer that we read into in the second question is printed out onto the terminal.

Altogether

If we look at these three questions together, we can do something with this program. If we give the inputs "flag.txt" and 0 for the first question, we will be opening the flag.txt file with a file descriptor of 3. For the second question, if we give an input of 5, then the program will end up with a 3 after manipulating our input, which it will then use as a file descriptor. We know that 3 is the file descriptor of falg.txt. Therefore, we can use this file descriptor to read the contents of flag.txt into a variable (let's call it some_string). For the third question, if we give an input of 2 using pwntools, then we can print out the contents of some_string to the terminal. Thus, we obtained the flag and the challenge is completed.