1. tool used: Running the File
    a. What we learned: The file deletes itself
2. Tool used: Strings
    a. What the tool does: prints out to the terminal all of the printable characters in the file, i.e., ascii characters
    b. What we learned:
        i. It seems that it is written in at least C, possibly with C++, because one of the lines printed is stderr
        ii. It does not delete all my files like your comment says >: (
        iii. There appears to be a rudimentary menu for debugging
            1. After finally figuring out how to run it through the command line, I realized that this menu is not for debugging but for what the program actually does
3. Tool used: File
    a. What the tool does: returns the details of said file
    b. What we learned:
        i. The file is an 64-bit ELF executable
4. Tool used: objdump –d
    a. What the tool does: displays information from an object
5. Tool used: nm mystery –C
    a. What the tool does: nm outputs the symbol table of the file and –C turns it into readable text
    b. What we learned:
        i. I was able to read all of the function calls of the c program without the parameters.

What the file does

The file prints the string If run through the command line without any flags: “I have deleted all your files …" then using sleep, sleeps for an amount of time before it prints “not”. Then it deletes itself. Using the –h flag will bring up a menu that then does not delete itself. Using the -n flag plus a number will create that many numbers and store them. Using the –e flag plus number will create a whole bunch of random numbers. Using the –s will sort the numbers that have either been allocated from -I or generated from –e. The –p flag supposedly sends data out

I think that the bug is that it deletes itself when a flag is not passed.