Author: Sinai Park(sp46)
Sebene Yi (ssy3)
Date: March 30, 2021
For CS232, Spring2021

Tool used: ./mystery
What the tool does: The command line runs the program
What we learned: The program deletes itself and writes to stderr

Tool used: strings
What the tool does: It is determining the contents of non-text file.
What we learned: We found strings in the file working as executables (stderr, strlen, atoi, close, accept, sleep, strcmp, srandom, malloc, stdout, strlen, sprintf, fflush, exit, puts, stack, so we know it's a C file)

Tool used: ls -l mystery (Terminal)
What the tool does: It lists information about the program
What we learned: The programs owner is sp46, the file group is cs-graders-21sp-cs232, the file size is 14008, and its modification time was Feb 29 of 2020, and the file's name is mystery.

Tool used: df -h mystery (Terminal)
What the tool does: The command shows amount of free space
What we learned: Its filesystem name is katz-nfs.cs.calvin.edu:/home and total size in m is 3.9T and it had used 575G and there are 3.1T available. It's used around 16% percent and it is mounted in the /home directory.

Tool used: file mystery (Terminal)
What the tool does: The command shows properties/compression of a file
What we learned: It is a ELF 64-bit LSB executable, x86-64, version 1, dynamically linked, and has a interpreter, for GNU/Linux 2.6.32, is not stripped

Tool used: stat mystery (Terminal)
What the tool does: The command outputs file status
What we learned: We learn that it has 32 blocks, IO blocks of 1048576, regular file, access: (0446/-rw-rw-r--)

Tool used: head mystery
What the tool does: The command shows first 10 lines
What we learned: It shows a little bit of human readable code along with encrypted text. (62; c)

Tool used: gdb ./mystery
What the tool does: The command allows the debugging of a c code file
What we learned: It shows that it has a little bit of human readable code along with encrypted text

Tool used: strace ./mystery p -3000
What the tool does: It traces all the outputs happening through the specified port
What we learned: We found that it monitors the instructions between processes and linux kernel

Tool used: readelf -all mystery

What the tool does: The command displays info for ELF files
What we learned: The entry point address is at 0x400a50

Tool used: objdump mystery
What the tool does: The command displays info for object files
What we learned: It gave the assembly code and shows a.out, little endian values, big endian values ; the code is compiled using GCC

Command Line
Mystery -h
What it does: This flag displays the manual help page for mystery.

Mystery -n
What it does: The command displays a specified number of random numbers

Mystery -p
What it does: listens in on an specific port instead of the default 10234
Can type nc localhost <port> on a separate terminal window to see the code and printing

Mystery -s
What it does: This command sorts the numbers from the smallest to greatest

Mystery -e
What it does: It seeds the random number generator; giving a same output with the same seed, but different numbers with a different value of seed

When the program just runs, it deletes itself upon execution. On the command line, it prints the words, “I just deleted all your files... not.” When the flags are in use, however, the command line running mystery shows something totally different. The -h shows the lists of the flags and what each one does. The mystery -n shows a number of random numbers of a specified value on each line. With the -p flag, we see that this gets connected on a localhost port, where the numbers are then separately written on a different file descriptor (after being connected to the socket). With -s, it sorts the random numbers and the -e shows a different, or unique generation of 100 numbers.

BUG:

- There is no default port initialized! If we don’t specify a port, we do not connect to a port, although it tells us that it will be connected to port 10234 by default.
- The <i> value after the n flag has to be a positive value for the program to run, it cannot be a negative value or a character. If it is specified as a negative number of a character, there’s no error that prints out and the program simply returns to the command line.

Command line linux source:
https://www.codementor.io/linux/tutorial/10-things-every-linux-beginner-should-know
https://www.codementor.io/@packt/reverse-engineering-a-linux-executable-hello-world-rjceryk5d