Aliyah Flores
aff13
Project 2

Password 1: bDZPslhIyfqDSFcWPErOAoVarxUTMj

Failed attempts: At first, I directly went in with gdb and tried to work through the program. Instead, this felt too time-consuming and I felt like there was a quicker solution.

Ideas: While reading tips given on the project description, I thought about using the mystrings program that I had written to see if the password can appear as a string when hexdumped.

Solution: I used mystrings and sifted "ctr+f" the words "Congratulations…," since I knew that the password would appear if I had gotten the success message. When I found the the congratulatory message, there was a string directly above from it. I copy-pasted and ran the first executable with it and found that it was a success.

Password 2: 24.229.207.16 51685 22

Failed attempts: Because I was hoping that the second password would be as simple as the first one (I was sorely disappointed), I tried the mystrings method once again. While doing this, I plugged in some of the strings that I assumed could be the password but none of them worked.

Ideas: I didn't know whether to go write in with gdb or whether to try out the other methods given in the project instructions. So I decided to try each of them out; first I used nm, then hexdump (both which didn't seem to help me), and finally objdump. I was overwhelmed with the options and ultimately decided to use gdb instead.

Solution: What I did first was put a breakpoint in the main function. After I did this, I kind of just sifted through all the steps and examined what was in each register. Very time-consuming. I didn't find anything, so I got stumped for a while. After this, I tried to see what other functions there were in the executable. Since there were a few, I set breakpoints at the start of each function and did the same thing I did with main… which was stepping through each method instruction and examining the registers. Pretty soon, I found content that didn't seem to match the other gibberish I was seeing and copy-pasted it. After trying it out, the attempt seemed to work and I had found the second password.

Password 3: [Not found]

Failed attempts: There were… a lot of failed attempts. This took me at around 3-5 days of tinkering before deciding to give up (although I feel like I could have figured it out if the deadline hadn't been so close). First, I tried to use mystrings like I did in executable one, but all the strings didn't work when I tried them as passwords. Next, I attempt to use gdb but found that the main() function was undefined (yikes).

I tried to use hexdump to locate the strings that way, but the results were still encrypted so I knew I had to dig in deep. Instead, I decided to use objdump (even though it terrified me) and I used objdump –S aff13_3 to view all of the source code. Since there was a lot, I narrowed it down to the .text section. Instead of finding a main() function, I saw <__libc_start_main@plt> and Googled it to find that this meant the

executable was dynamically loaded. This was where I decided to put my first breakpoint. It was a start. I tried to do what I did for the second password and sifted through the instructions for hours upon hours, but nothing seemed to work.

I then tried to trace the code, but I had difficulty following where which value was put into which register and so on. This attempt was futile and took up a whole day of coding. Eventually, I realized that I probably couldn't get the password, so I tried to Google some different approaches.

Ideas: After a LOT of researching, learning, and experimentation, I could safely say that I got close to cracking the password, but still couldn't figure it out in the end. As the deadline is quickly approaching, one of the ways I wanted to try but didn't have time to was this:

Using objdump and hexdump, I found the corresponding hex values for the "Congratulations" and "Sorry!" messages. I tried to find these in the .text portion of the executable and found them, but when inspecting the registers, could not find any values that looked like a password. I then traced the code further up from the congratulations message to see where I could put a breakpoint. I did this in multiple spots and got close, where I learned that there had to be 15 values in the password. What those values were, I didn't get to find out. But I hope I was somewhere along the right lines.

Solution: n/a