Adam Granieri

COE 0449 Systems Software

Project 3 Report

Dr. Lange

adg88\_1

Solution Passphrase: vnAWqPycLOpFMnNVdnAYqI

Methodology:

I first ran the program and entered in a random series of characters for a password to see what it would return. I saw it displayed “Sorry! Not Correct” and took note of that. I then made the mystrings program to copy the strings program on all three executables. After making the mystrings.c program I utilized it by running on “adg88\_1” and directing the output to a file called “output”. From there I opened “output” in Nano and utilized the “where is” function and searched for “sorry”. It brought me to a line with the incorrect output text and the correct output text. I saw a long string of characters right above that and thought to try the password. I exited and ran the executable and input the above password and saw it passed.

**adg88\_2**

Solution Passphrase: aaaaaaaaaaaaaaaaa or any 14-character or longer string that is also a palindrome

Methodology:

After running mystrings on the executable I didn’t see any strings that were encouraging to try. Instead I used nm on the executable and looked for the symbols of any potential keys and functions. I found main, c, s, and p. I then ran gdb on the file and used disas on main. I saw that main goes an fgets on the input and then calls c and s. I ran disas on c and saw it seem to just be looping through some data checking if things were unequal then branching to potentially call s. I then disassembled s to see it just checked a few things on the stack, but I couldn’t really tell. Lastly, I just checked p just for clarity and it was just printing out the end result. My guess was that the password wasn’t any singular word, but some set of criteria. For the heck of it I tried entering aaaaaaaaaaaaaaaaa and it worked. My intuition was to see if I could enter in a longer string, so I entered aaaaaaaaaaaaaaaaaaa which also worked. I continually entered in the string with one less character until it rejected it and I found the count was 14 minimum. With a correct password I went in to see what was happening. I set breakpoints at the first call to c in main and in the comparison line in c. I saw it was checking the beginning and end of the input and checking if they were equal then incrementing the beginning pointer and decrementing the end pointer. This continued until the pointers equaled or pasted each other. When I realized the comparison, I checked to see if anything checked for length of input and saw what seemed to be a check in s.

**adg88\_3**

Solution Passphrase: 123456789123456 or any rotation of numerically increasing sequence

Methodology:

I first ran strings to see if anything helpful would show up, but nothing did. I then tried to find the symbols in the file, but it was stripped, so I got nothing then either. I then tried to run disas on main, but there was no main in the function which meant that it was loading a main function from somewhere else. I didn’t know what to do, so distraught, I ran the executable and put in 123456789, but then when I hit enter it gave me a new line, so I typed in 123456 and it worked. From this I found that the password could take multiple lines. So, I tried entering the same string without separating lines and that worked as well. My other attempts doing a similar attempt of increasing inputs failed with every case of using letters. I did get 234567892345 to work as well which lead me to believe that any rotation of a the initially correct sequence would work and passed my test cases of starting any number and then increasing single digit values. Ultimately I could not determine what the program was doing, but I was at a loss of how to analyze the program execution. This executable was very odd.