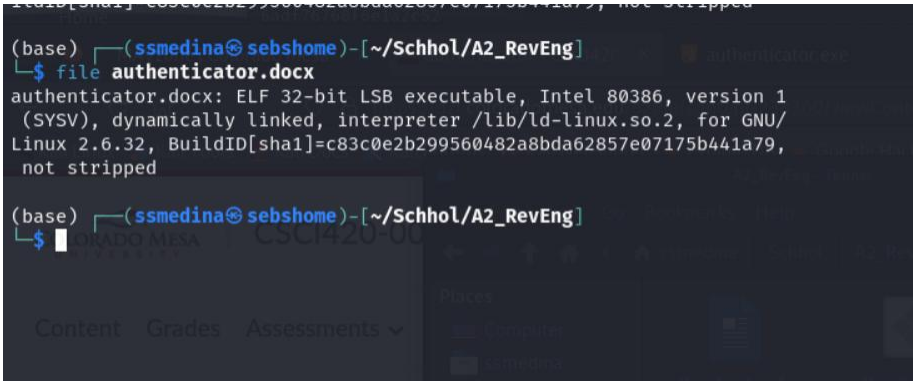
Reverse Engineering Assignment

By Sebastian Medina

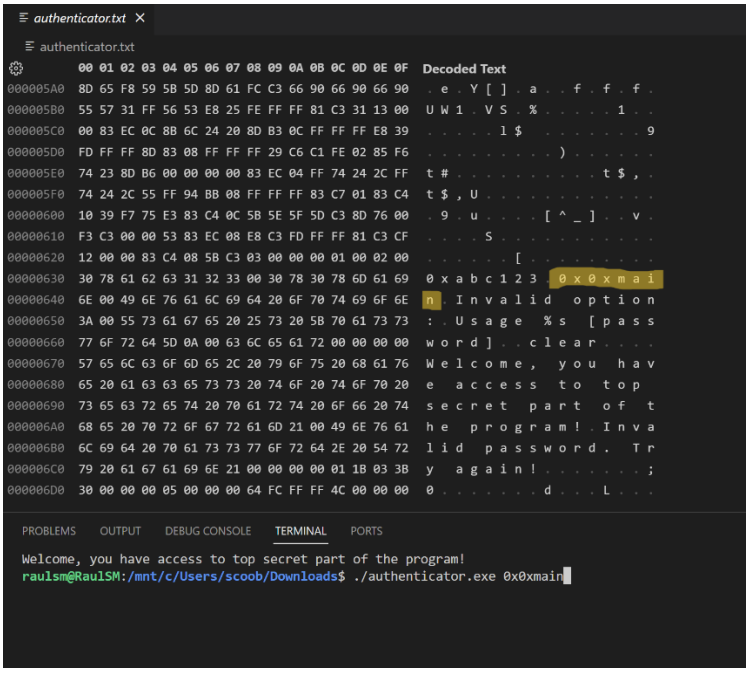
Content:

Report of Reverse Engineering:

To start this assignment, I had to determine what the file type of authenticator.docx was because when I downloaded it was just a .docx file. The command I used to determine the file type is “file authenticator.docx”. This allowed me to see that authenticator.docx is an ELF 32-bit executable file as shown by the screenshot below. I then changed authenticator.docx to authenticator.exe to make it a executable.

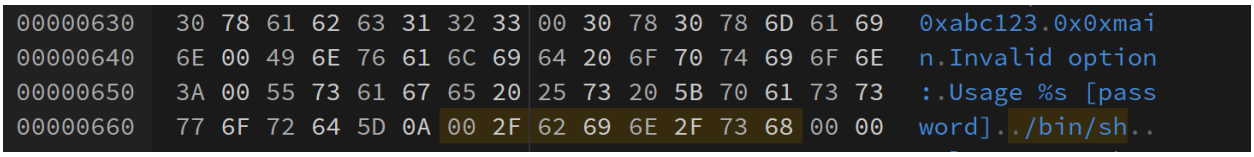


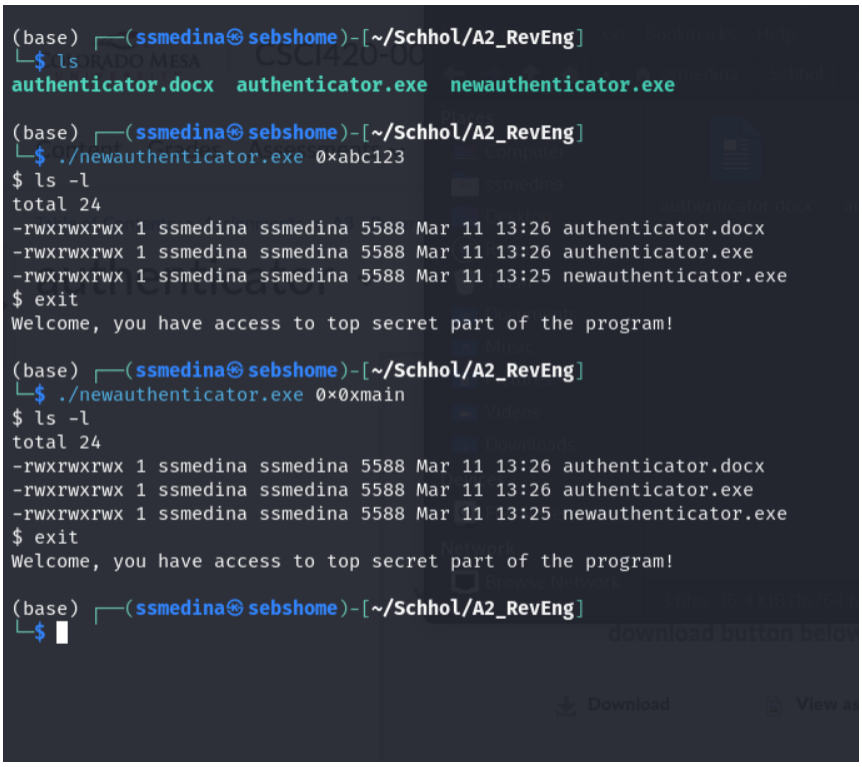
The next part of this assignment I had to find two default passwords to enter when I ran the program to get the message “Welcome, you have access to the top-secret part of the program!”. To find these passwords, I used a hex editor to examine the binary code of authenticator.exe and looked for were they talked about a welcome screen and inputting a password. When I found this section, I noticed two lines at the beginning of this section, one spelling out 0xabc123 and the other saying 0x0xmain. I thought it was weird that they were in this section, so I entered them for the password and they both worked and displayed the message I was supposed to see shown in the snapshot below.



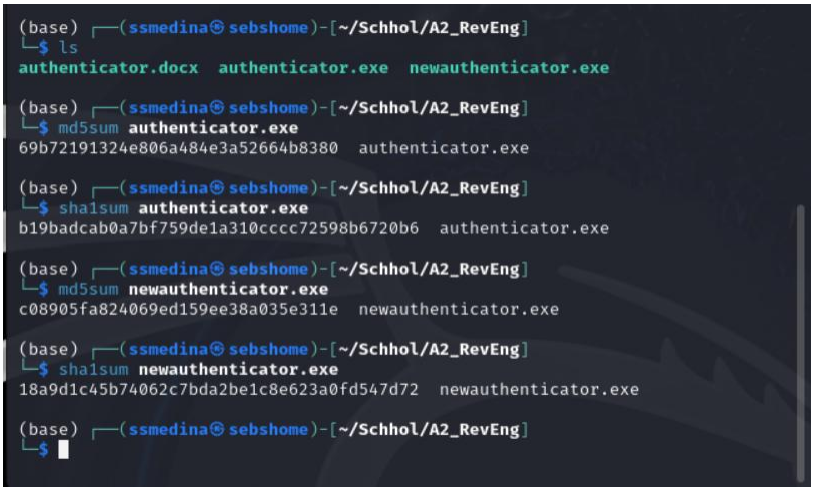
For the second to last step for this assignment I had to modify the binary so that when I input the correct password for the program the program would execute /bin/sh shell. To do this, I looked up what /bin/sh is in hex form is which is 2f 62 69 6e 2f 73 68. Once I knew this, I looked though the binary though a hex editor and looked for a place I can enter the /bin/sh so that the program would recognize

and run it. While I was looking though the hex editor, I notice a that there was a line that said clear after you enter the correct password meaning that it was doing nothing after you entered the correct password. So, I edit this section of the binary file with a hex editor where I replaced clear with /bin/sh hex form and ran the program to see what happen. I save the changes to a new .exe file named newathenticator.exe. When I ran the program with the correct password with this edit it executed /bin/sh were I entered the shell and then when I exited the shell the message “Welcome, you have access to the top-secret part of the program!” printed and then the program ended. Below is all the snapshots I toke about the step above.





For the lase step of this assignment I had to research checksums, md5, and sha1. Then I had to calculate md5 and sha1 of the modified and original .exe files. Checksums are the type of value calculated in a set of data so that it can detect error. They work by applying a mathematical algorithm to the data set. Another fun fact about checksum is that when data is transmitted or stored, the checksum is also sent with the data. Sha1 and md5 are both very popular checksum algorithms that are used to calculate the checksum. To calculate the checksum with md5 the command you use is md5sum <filename> and to calculate the checksum with sha1 the command you use is sha1sum <filename>. I used these commands for both .exe files to find the checksum of both files. The examples of the command being run as well as the calculated checksum for both .exe files in md5 and sha1 are shown in the snapshot below.



This is how I completed this assignment and the steps I toke in order to complete each action this assignment wanted me to complete.