

Reverse Engineering Assignment

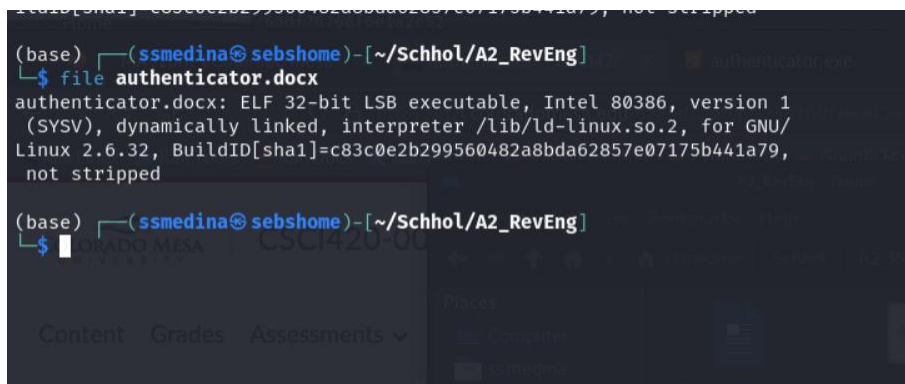
By Sebastian Medina

Content:

Report of Reverse Engineering.....pg1

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.

A terminal window with a dark background. The prompt is (base) and the user is ssmedina@sebshome in the directory ~/Schhol/A2_RevEng. The command 'file authenticator.docx' has been entered. The output shows that authenticator.docx is an ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.32, with a SHA1 hash of c83c0e2b299560482a8bda62857e07175b441a79, and it is not stripped. The prompt is now \$.

```
(base) (ssmedina@sebshome) - [~/Schhol/A2_RevEng]
$ file authenticator.docx
authenticator.docx: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/
Linux 2.6.32, BuildID[sha1]=c83c0e2b299560482a8bda62857e07175b441a79,
not stripped

(base) (ssmedina@sebshome) - [~/Schhol/A2_RevEng]
$
```

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 where 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.

```
authenticator.txt X
C:\Users\scoob\Downloads\authenticator.txt
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded Text
000005A0 8D 65 F8 59 5B 5D 8D 61 FC C3 66 90 66 90 66 90 .e.Y[ ].a.f.f.f.
000005B0 55 57 31 FF 56 53 E8 25 FE FF FF 81 C3 31 13 00 UW1.VS.%...1..
000005C0 00 83 EC 0C 8B 6C 24 20 8D B3 0C FF FF FF E8 39 .....l$......9
000005D0 FD FF FF 8D 83 08 FF FF FF 29 C6 C1 FE 02 85 F6 .....).t...t$,.
000005E0 74 23 8D B6 00 00 00 83 EC 04 FF 74 24 2C FF t#.....t$,.
000005F0 74 24 2C 55 FF 94 BB 08 FF FF FF 83 C7 01 83 C4 t$,U.....
00000600 10 39 F7 75 E3 83 C4 0C 5B 5E 5F 5D C3 8D 76 00 .9.u.....[^_].v.
00000610 F3 C3 00 00 53 83 EC 08 E8 C3 FD FF FF 81 C3 CF .....S.....
00000620 12 00 00 83 C4 08 5B C3 03 00 00 00 01 00 02 00 .....[.....
00000630 30 78 61 62 63 31 32 33 00 30 78 30 78 6D 61 69 0xabc123 0x0xmai
00000640 6E 00 49 6E 76 61 6C 69 64 20 6F 70 74 69 6F 6E n.Invalid option
00000650 3A 00 55 73 61 67 65 20 25 73 20 5B 70 61 73 73 :.Usage %s [pass
00000660 77 6F 72 64 5D 0A 00 63 6C 65 61 72 00 00 00 00 word].clear...
00000670 57 65 6C 63 6F 6D 65 2C 20 79 6F 75 20 68 61 76 Welcome, you hav
00000680 65 20 61 63 63 65 73 73 20 74 6F 20 74 6F 70 20 e access to top
00000690 73 65 63 72 65 74 20 70 61 72 74 20 6F 66 20 74 secret part of t
000006A0 68 65 20 70 72 6F 67 72 61 6D 21 00 49 6E 76 61 he program!.Inva
000006B0 6C 69 64 20 70 61 73 73 77 6F 72 64 2E 20 54 72 lid password. Tr
000006C0 79 20 61 67 61 69 6E 21 00 00 00 01 1B 03 3B y again!.....;
000006D0 30 00 00 00 05 00 00 00 64 FC FF FF 4C 00 00 00 0 .d...L...

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Welcome, you have access to top secret part of the program!
raulsm@raulsm:/mnt/c/Users/scoob/Downloads$ ./authenticator.exe 0xabc123
```

```
authenticator.txt X
authenticator.txt
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded Text
000005A0 8D 65 F8 59 5B 5D 8D 61 FC C3 66 90 66 90 66 90 .e.Y[ ].a.f.f.f.
000005B0 55 57 31 FF 56 53 E8 25 FE FF FF 81 C3 31 13 00 UW1.VS.%...1..
000005C0 00 83 EC 0C 8B 6C 24 20 8D B3 0C FF FF FF E8 39 .....l$......9
000005D0 FD FF FF 8D 83 08 FF FF FF 29 C6 C1 FE 02 85 F6 .....).t...t$,.
000005E0 74 23 8D B6 00 00 00 83 EC 04 FF 74 24 2C FF t#.....t$,.
000005F0 74 24 2C 55 FF 94 BB 08 FF FF FF 83 C7 01 83 C4 t$,U.....
00000600 10 39 F7 75 E3 83 C4 0C 5B 5E 5F 5D C3 8D 76 00 .9.u.....[^_].v.
00000610 F3 C3 00 00 53 83 EC 08 E8 C3 FD FF FF 81 C3 CF .....S.....
00000620 12 00 00 83 C4 08 5B C3 03 00 00 00 01 00 02 00 .....[.....
00000630 30 78 61 62 63 31 32 33 00 30 78 30 78 6D 61 69 0xabc123 0x0xmai
00000640 6E 00 49 6E 76 61 6C 69 64 20 6F 70 74 69 6F 6E n.Invalid option
00000650 3A 00 55 73 61 67 65 20 25 73 20 5B 70 61 73 73 :.Usage %s [pass
00000660 77 6F 72 64 5D 0A 00 63 6C 65 61 72 00 00 00 00 word].clear...
00000670 57 65 6C 63 6F 6D 65 2C 20 79 6F 75 20 68 61 76 Welcome, you hav
00000680 65 20 61 63 63 65 73 73 20 74 6F 20 74 6F 70 20 e access to top
00000690 73 65 63 72 65 74 20 70 61 72 74 20 6F 66 20 74 secret part of t
000006A0 68 65 20 70 72 6F 67 72 61 6D 21 00 49 6E 76 61 he program!.Inva
000006B0 6C 69 64 20 70 61 73 73 77 6F 72 64 2E 20 54 72 lid password. Tr
000006C0 79 20 61 67 61 69 6E 21 00 00 00 01 1B 03 3B y again!.....;
000006D0 30 00 00 00 05 00 00 00 64 FC FF FF 4C 00 00 00 0 .d...L...

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Welcome, you have access to top secret part of the program!
raulsm@raulsm:/mnt/c/Users/scoob/Downloads$ ./authenticator.exe 0x0xmain
```

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 through the hex editor, I noticed 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 edited this section of the binary file with a hex editor where I replaced clear with /bin/sh in hex form and ran the program to see what happens. I saved the changes to a new .exe file named newauthenticator.exe. When I ran the program with the correct password with this edit it executed /bin/sh where 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 are all the snapshots I took about the step above.

```
00000660  77 6F 72 64 5D 0A 00 63 6C 65 61 72 00 00 00 00  word]..clear....
```

```
00000630  30 78 61 62 63 31 32 33 00 30 78 30 78 6D 61 69  0xabc123.0x0xmai
00000640  6E 00 49 6E 76 61 6C 69 64 20 6F 70 74 69 6F 6E  n.Invalid option
00000650  3A 00 55 73 61 67 65 20 25 73 20 5B 70 61 73 73  :.Usage %s [pass
00000660  77 6F 72 64 5D 0A 00 2F 62 69 6E 2F 73 68 00 00  word]../bin/sh..
```

```
(base) (ssmedina@sebshome) ~/Schhol/A2_RevEng
$ ls
authenticator.docx  authenticator.exe  newauthenticator.exe

(base) (ssmedina@sebshome) ~/Schhol/A2_RevEng
$ ./newauthenticator.exe 0xabc123
$ ls -l
total 24
-rwxrwxrwx 1 ssmedina ssmedina 5588 Mar 11 13:26 authenticator.docx
-rwxrwxrwx 1 ssmedina ssmedina 5588 Mar 11 13:26 authenticator.exe
-rwxrwxrwx 1 ssmedina ssmedina 5588 Mar 11 13:25 newauthenticator.exe
$ exit
Welcome, you have access to top secret part of the program!

(base) (ssmedina@sebshome) ~/Schhol/A2_RevEng
$ ./newauthenticator.exe 0x0xmain
$ ls -l
total 24
-rwxrwxrwx 1 ssmedina ssmedina 5588 Mar 11 13:26 authenticator.docx
-rwxrwxrwx 1 ssmedina ssmedina 5588 Mar 11 13:26 authenticator.exe
-rwxrwxrwx 1 ssmedina ssmedina 5588 Mar 11 13:25 newauthenticator.exe
$ exit
Welcome, you have access to top secret part of the program!

(base) (ssmedina@sebshome) ~/Schhol/A2_RevEng
$
```

For the last 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.

```
(base) └─(ssmedina@sebshome)-[~/Schhol/A2_RevEng]
└─$ ls
authenticator.docx  authenticator.exe  newauthenticator.exe

(base) └─(ssmedina@sebshome)-[~/Schhol/A2_RevEng]
└─$ md5sum authenticator.exe
69b72191324e806a484e3a52664b8380  authenticator.exe

(base) └─(ssmedina@sebshome)-[~/Schhol/A2_RevEng]
└─$ sha1sum authenticator.exe
b19badcab0a7bf759de1a310cccc72598b6720b6  authenticator.exe

(base) └─(ssmedina@sebshome)-[~/Schhol/A2_RevEng]
└─$ md5sum newauthenticator.exe
c08905fa824069ed159ee38a035e311e  newauthenticator.exe

(base) └─(ssmedina@sebshome)-[~/Schhol/A2_RevEng]
└─$ sha1sum newauthenticator.exe
18a9d1c45b74062c7bda2be1c8e623a0fd547d72  newauthenticator.exe

(base) └─(ssmedina@sebshome)-[~/Schhol/A2_RevEng]
└─$
```

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