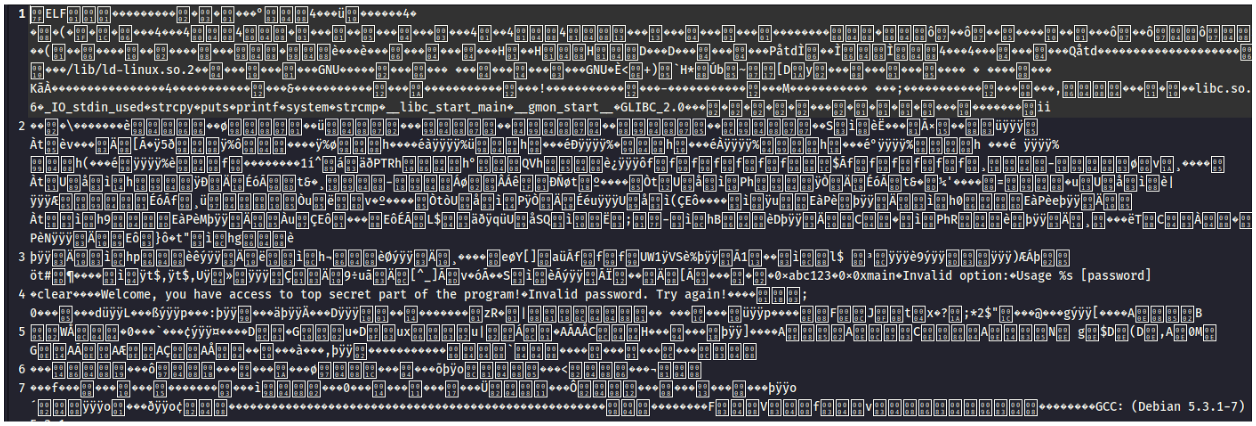
Reverse Engineering
Bailey Williams

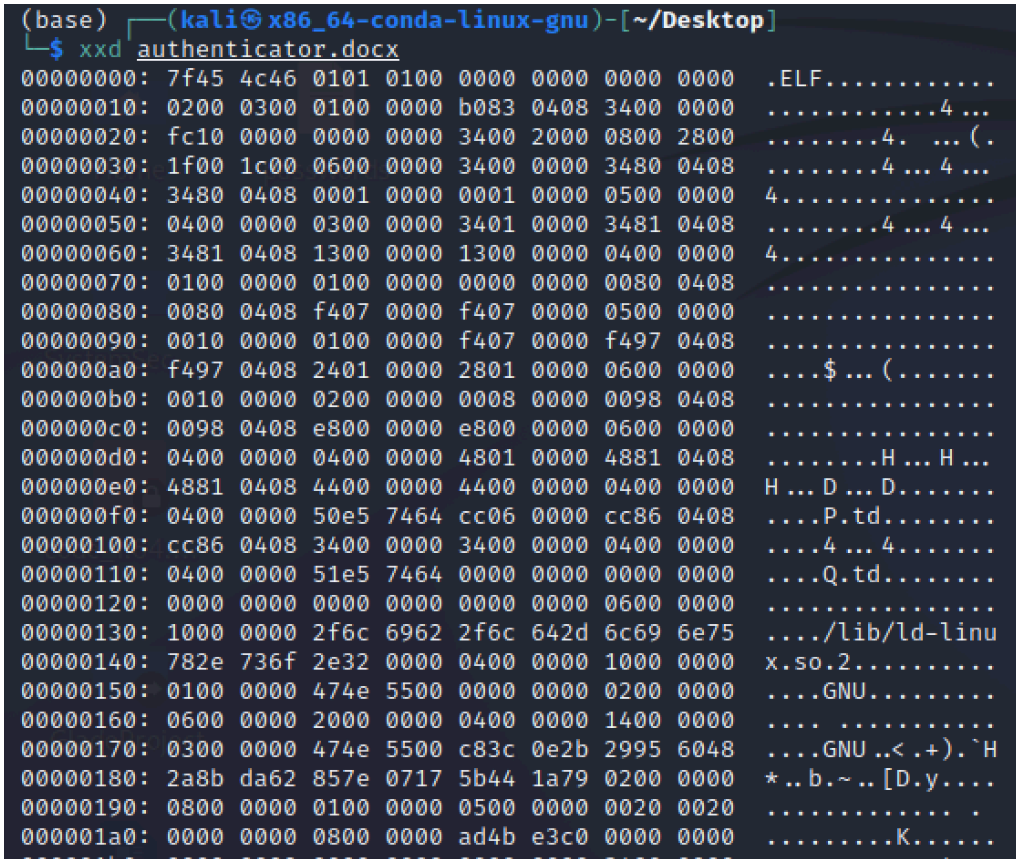
1. Determine the type/format of the authenticator.docx file.

- By putting the authenticator file in Mousepad it shows the Executable and Linkable Format (ELF).



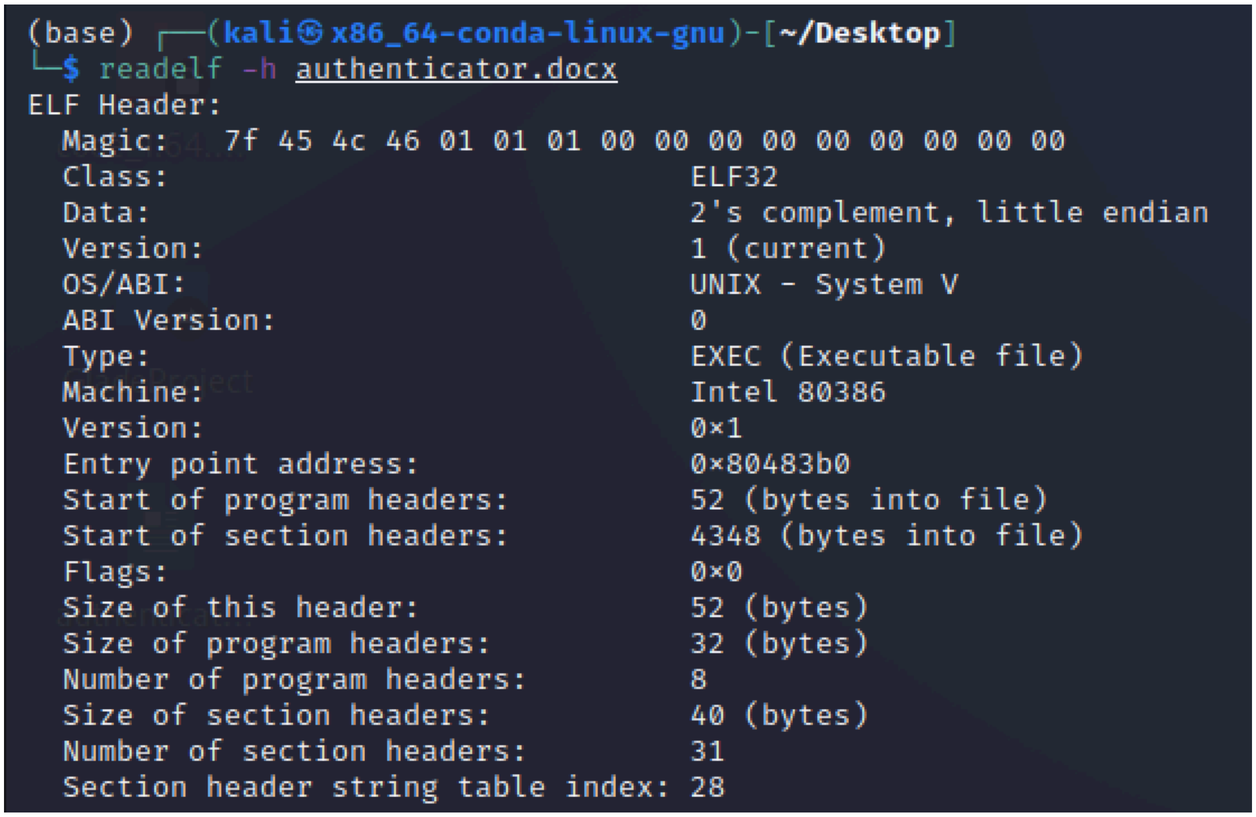
```
1 [illegible]ELF[illegible]4[illegible]4
  [illegible]4[illegible]4[illegible]4[illegible]PåtdÌ[illegible]Qåtd[illegible]
  [illegible]/lib/ld-linux.so.2[illegible]GNU[illegible]GNU•È<[illegible]+)[illegible]`H*[illegible]Úb[illegible]~[illegible][D[illegible]y[illegible]
  KãÀ[illegible]libc.so.
  6•_IO_stdin_used•strcpy•puts•printf•system•strcmp•__libc_start_main•__gmon_start__•GLIBC_2.0[illegible]ii
2 [illegible]
  [illegible]
3 [illegible]
  [illegible]•0×abc123•0×0xmain•Invalid option:•Usage %s [password]
4 •clear••••Welcome, you have access to top secret part of the program!•Invalid password. Try again!••••[illegible];
  [illegible]
5 [illegible]
6 [illegible]
7 [illegible]
  [illegible]GCC: (Debian 5.3.1-7)
```

- By using xxd on the authenticator file it also shows the ELF format.



```
(base) ┌──(kali㉿x86_64-conda-linux-gnu)-[~/Desktop]
└─$ xxd authenticator.docx
00000000: 7f45 4c46 0101 0100 0000 0000 0000 0000  .ELF............
00000010: 0200 0300 0100 0000 b083 0408 3400 0000  ............4...
00000020: fc10 0000 0000 0000 3400 2000 0800 2800  ........4. ...(.
00000030: 1f00 1c00 0600 0000 3400 0000 3480 0408  ........4...4...
00000040: 3480 0408 0001 0000 0001 0000 0500 0000  4...............
00000050: 0400 0000 0300 0000 3401 0000 3481 0408  ........4...4...
00000060: 3481 0408 1300 0000 1300 0000 0400 0000  4...............
00000070: 0100 0000 0100 0000 0000 0000 0080 0408  ................
00000080: 0080 0408 f407 0000 f407 0000 0500 0000  ................
00000090: 0010 0000 0100 0000 f407 0000 f497 0408  ................
000000a0: f497 0408 2401 0000 2801 0000 0600 0000  ....$...(.......
000000b0: 0010 0000 0200 0000 0008 0000 0098 0408  ................
000000c0: 0098 0408 e800 0000 e800 0000 0600 0000  ................
000000d0: 0400 0000 0400 0000 4801 0000 4881 0408  ........H...H...
000000e0: 4881 0408 4400 0000 4400 0000 0400 0000  H...D...D.......
000000f0: 0400 0000 50e5 7464 cc06 0000 cc86 0408  ....P.td........
00000100: cc86 0408 3400 0000 3400 0000 0400 0000  ....4...4.......
00000110: 0400 0000 51e5 7464 0000 0000 0000 0000  ....Q.td........
00000120: 0000 0000 0000 0000 0000 0000 0600 0000  ................
00000130: 1000 0000 2f6c 6962 2f6c 642d 6c69 6e75  ..../lib/ld-linu
00000140: 782e 736f 2e32 0000 0400 0000 1000 0000  x.so.2..........
00000150: 0100 0000 474e 5500 0000 0000 0200 0000  ....GNU.........
00000160: 0600 0000 2000 0000 0400 0000 1400 0000  .... ...........
00000170: 0300 0000 474e 5500 c83c 0e2b 2995 6048  ....GNU..<.+).`H
00000180: 2a8b da62 857e 0717 5b44 1a79 0200 0000  *..b.~..[D.y....
00000190: 0800 0000 0100 0000 0500 0000 0020 0020  ............. .
000001a0: 0000 0000 0800 0000 ad4b e3c0 0000 0000  .........K......
```

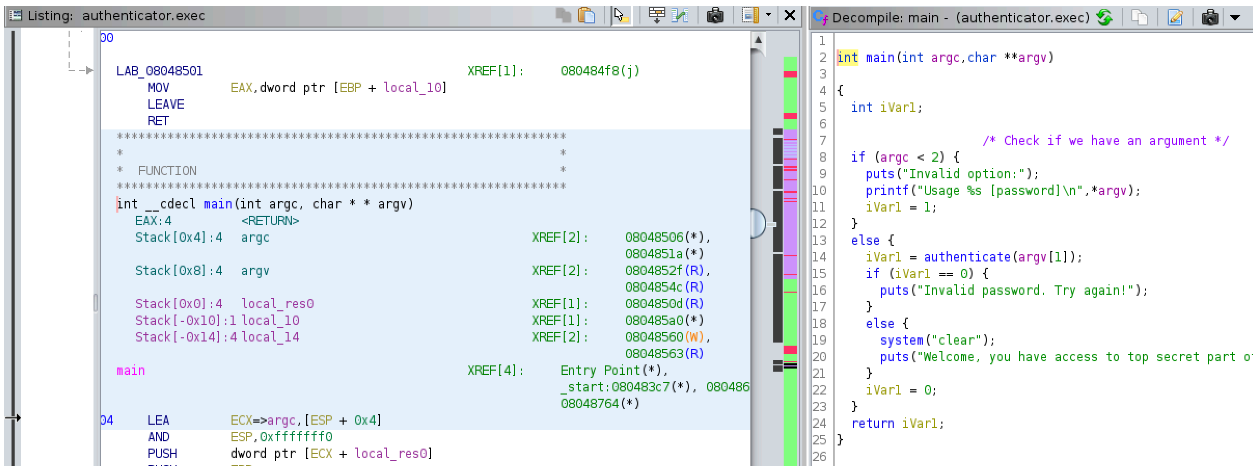
- By using the command *"readelf -h authenticator.docx"* it shows its as being an ELF32 Executable file.



```
(base) ┌──(kali㉿x86_64-conda-linux-gnu)-[~/Desktop]
└─$ readelf -h authenticator.docx
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x80483b0
  Start of program headers:          52 (bytes into file)
  Start of section headers:          4348 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         8
  Size of section headers:           40 (bytes)
  Number of section headers:         31
  Section header string table index: 28
```

2. Reverse Engineering the file.

- I used Ghidra to help organize and complete the reverse engineering.
- I first found the *main* function and analyzed it. I found that it called the function *authenticate*.

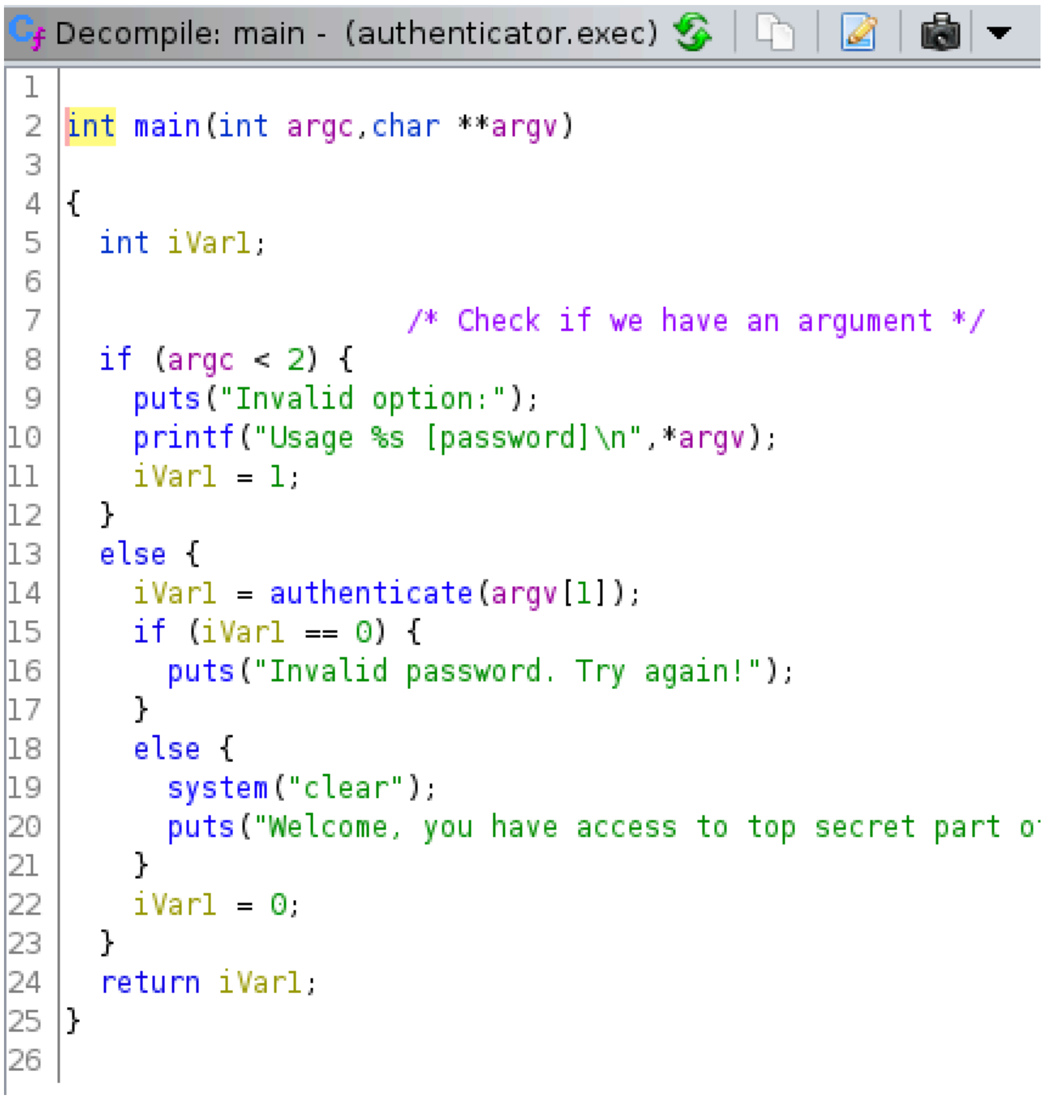


Listing: authenticator.exec

```
00

         LAB_08048501                              XREF[1]:     080484f8(j)
    MOV         EAX,dword ptr [EBP + local_10]
    LEAVE
    RET
  **************************************************************
  *                                                            *
  *  FUNCTION                                                  *
  **************************************************************
  int __cdecl main(int argc, char * * argv)
     EAX:4          <RETURN>
     Stack[0x4]:4   argc                           XREF[2]:     08048506(*),
                                                                0804851a(*)
     Stack[0x8]:4   argv                           XREF[2]:     0804852f(R),
                                                                0804854c(R)
     Stack[0x0]:4   local_res0                     XREF[1]:     0804850d(R)
     Stack[-0x10]:1 local_10                       XREF[1]:     080485a0(*)
     Stack[-0x14]:4 local_14                       XREF[2]:     08048560(W),
                                                                08048563(R)
  main                                     XREF[4]:     Entry Point(*),
                                                        _start:080483c7(*), 080486
                                                        08048764(*)
04  LEA         ECX=>argc,[ESP + 0x4]
    AND         ESP,0xfffffff0
    PUSH        dword ptr [ECX + local_res0]
```

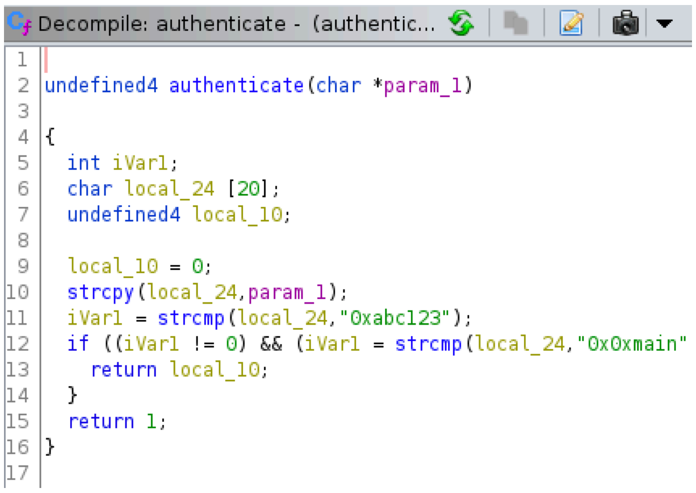
Decompile: main - (authenticator.exec)

```c

int main(int argc,char **argv)

{
  int iVar1;

                    /* Check if we have an argument */
  if (argc < 2) {
    puts("Invalid option:");
    printf("Usage %s [password]\n",*argv);
    iVar1 = 1;
  }
  else {
    iVar1 = authenticate(argv[1]);
    if (iVar1 == 0) {
      puts("Invalid password. Try again!");
    }
    else {
      system("clear");
      puts("Welcome, you have access to top secret part o
    }
    iVar1 = 0;
  }
  return iVar1;
}
```



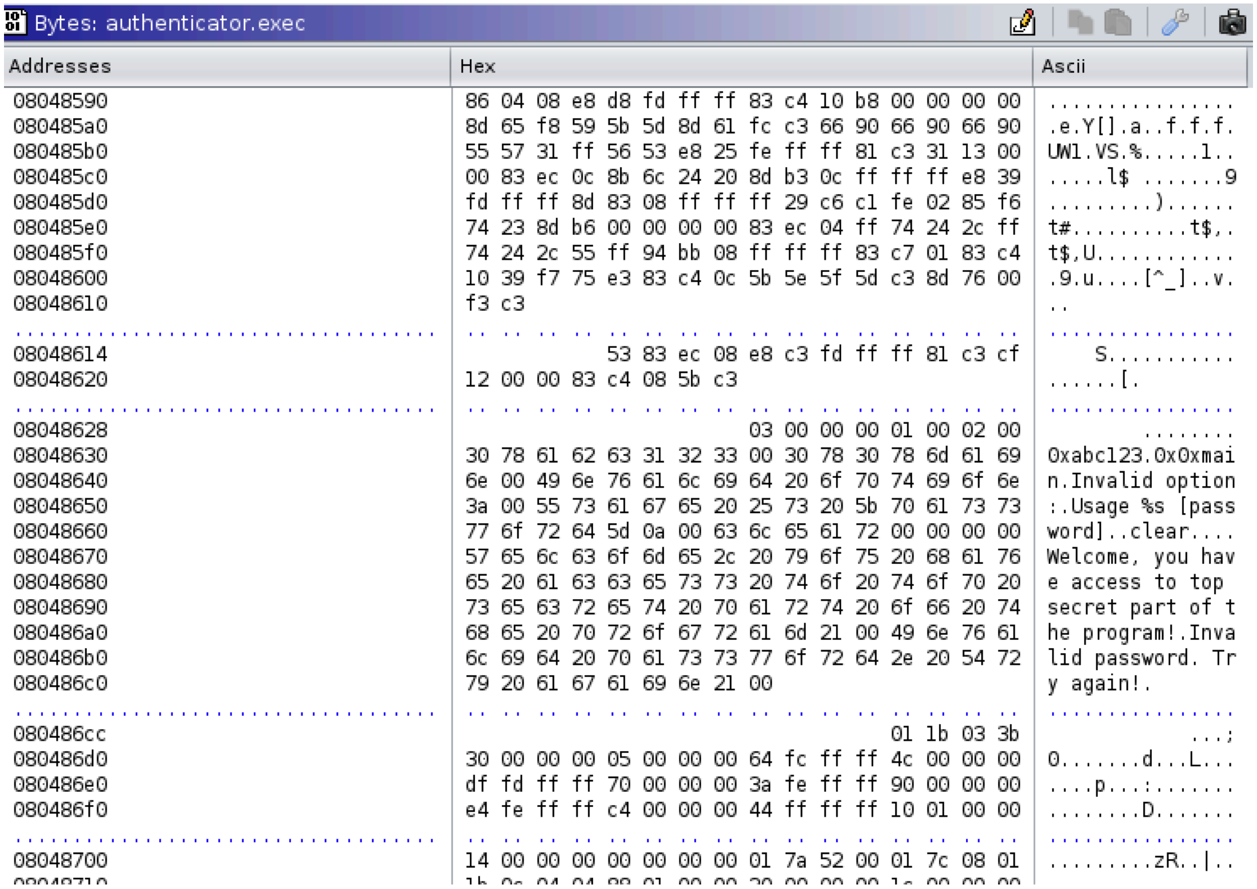
```
int main(int argc,char **argv)

{
  int iVar1;

                    /* Check if we have an argument */
  if (argc < 2) {
    puts("Invalid option:");
    printf("Usage %s [password]\n",*argv);
    iVar1 = 1;
  }
  else {
    iVar1 = authenticate(argv[1]);
    if (iVar1 == 0) {
      puts("Invalid password. Try again!");
    }
    else {
      system("clear");
      puts("Welcome, you have access to top secret part o
    }
    iVar1 = 0;
  }
  return iVar1;
}
```

- In the *authenticate* function I found that the user input was being compared to the correct passwords:
  - *0xabc123*
  - *0x0xmain*



```
undefined4 authenticate(char *param_1)

{
  int iVar1;
  char local_24 [20];
  undefined4 local_10;

  local_10 = 0;
  strcpy(local_24,param_1);
  iVar1 = strcmp(local_24,"0xabc123");
  if ((iVar1 != 0) && (iVar1 = strcmp(local_24,"0x0xmain"
    return local_10;
  }
  return 1;
}
```

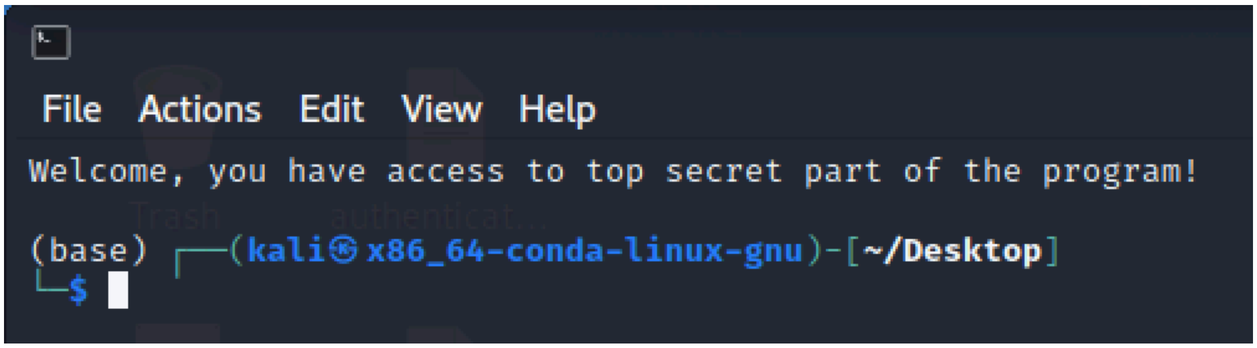
- I also found the same results looking through the ASCII.



```
Bytes: authenticator.exec

Addresses      Hex                                                Ascii
08048590       86 04 08 e8 d8 fd ff ff 83 c4 10 b8 00 00 00 00    ................
080485a0       8d 65 f8 59 5b 5d 8d 61 fc c3 66 90 66 90 66 90    .e.Y[].a..f.f.f.
080485b0       55 57 31 ff 56 53 e8 25 fe ff ff 81 c3 31 13 00    UW1.VS.%.....1..
080485c0       00 83 ec 0c 8b 6c 24 20 8d b3 0c ff ff ff e8 39    .....l$ .......9
080485d0       fd ff ff 8d 83 08 ff ff ff 29 c6 c1 fe 02 85 f6    .........)......
080485e0       74 23 8d b6 00 00 00 00 83 ec 04 ff 74 24 2c ff    t#..........t$,.
080485f0       74 24 2c 55 ff 94 bb 08 ff ff ff 83 c7 01 83 c4    t$,U............
08048600       10 39 f7 75 e3 83 c4 0c 5b 5e 5f 5d c3 8d 76 00    .9.u....[^_]..v.
08048610       f3 c3                                              ..
.................................................................................
08048614                   53 83 ec 08 e8 c3 fd ff ff 81 c3 cf        S...........
08048620       12 00 00 83 c4 08 5b c3                            ......[.
.................................................................................
08048628                               03 00 00 00 01 00 02 00            ........
08048630       30 78 61 62 63 31 32 33 00 30 78 30 78 6d 61 69    0xabc123.0x0xmai
08048640       6e 00 49 6e 76 61 6c 69 64 20 6f 70 74 69 6f 6e    n.Invalid option
08048650       3a 00 55 73 61 67 65 20 25 73 20 5b 70 61 73 73    :.Usage %s [pass
08048660       77 6f 72 64 5d 0a 00 63 6c 65 61 72 00 00 00 00    word]..clear....
08048670       57 65 6c 63 6f 6d 65 2c 20 79 6f 75 20 68 61 76    Welcome, you hav
08048680       65 20 61 63 63 65 73 73 20 74 6f 20 74 6f 70 20    e access to top 
08048690       73 65 63 72 65 74 20 70 61 72 74 20 6f 66 20 74    secret part of t
080486a0       68 65 20 70 72 6f 67 72 61 6d 21 00 49 6e 76 61    he program!.Inva
080486b0       6c 69 64 20 70 61 73 73 77 6f 72 64 2e 20 54 72    lid password. Tr
080486c0       79 20 61 67 61 69 6e 21 00                         y again!.
.................................................................................
080486cc                                           01 1b 03 3b                ...;
080486d0       30 00 00 00 05 00 00 00 64 fc ff ff 4c 00 00 00    0.......d...L...
080486e0       df fd ff ff 70 00 00 00 3a fe ff ff 90 00 00 00    ....p...:.......
080486f0       e4 fe ff ff c4 00 00 00 44 ff ff ff 10 01 00 00    ........D.......
.................................................................................
08048700       14 00 00 00 00 00 00 00 01 7a 52 00 01 7c 08 01    .........zR..|..
```

3. Using the passwords on the program.

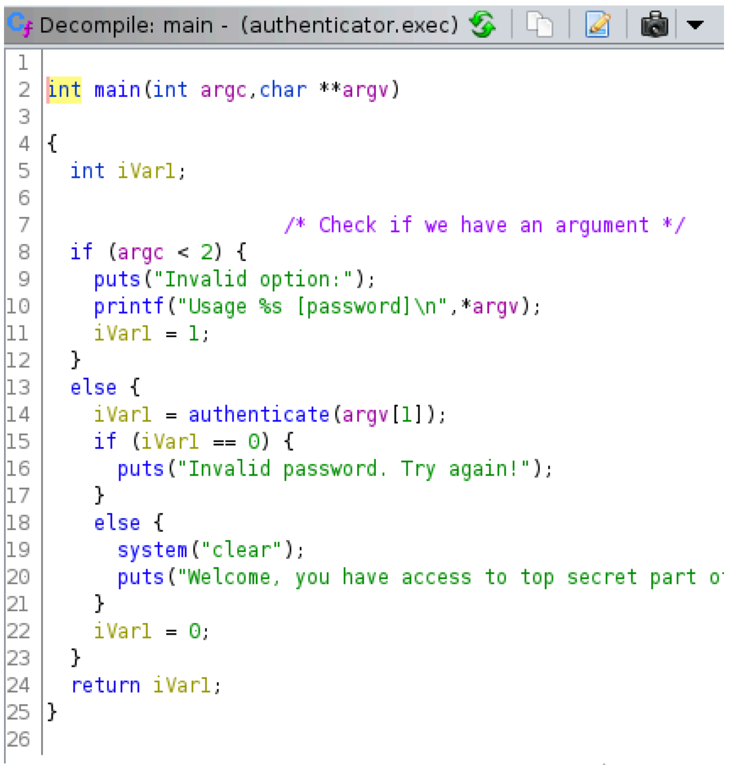
- Both passwords resulted in this message.



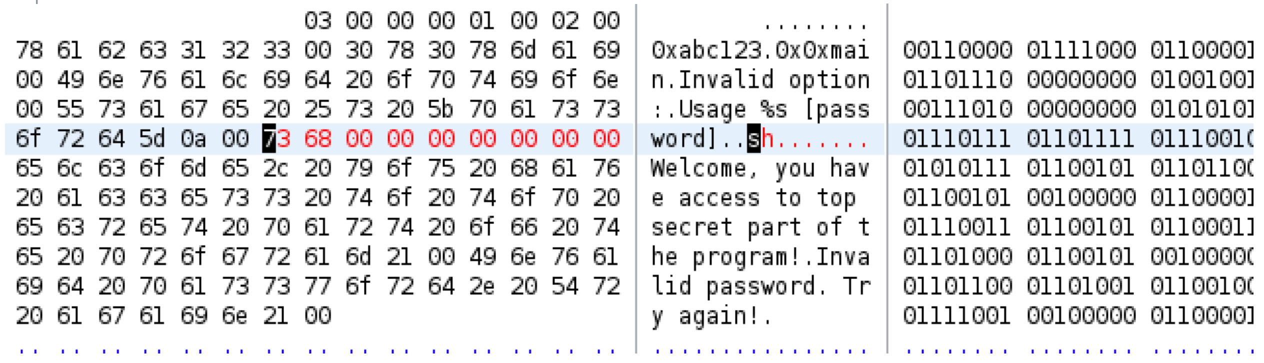
```
Welcome, you have access to top secret part of the program!

(base) ┌──(kali㉿x86_64-conda-linux-gnu)-[~/Desktop]
└─$ 
```

4. Modify the binary so that it executes /bin/sh shell program when the user uses the correct password.

- Using Ghidra I found in the *main* function *system("clear")* which I can modify to *system("sh")* so when a user puts in the correct password it will execute a shell program. With tools given by Ghidra, I modified the binary to change "clear" to "sh".

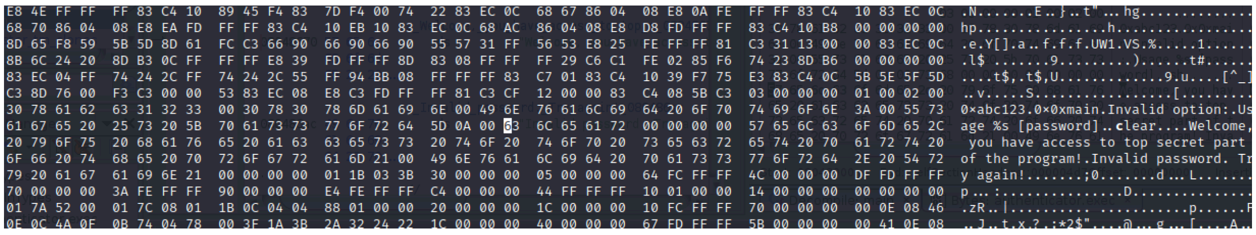


```
int main(int argc,char **argv)

{
  int iVar1;

                    /* Check if we have an argument */
  if (argc < 2) {
    puts("Invalid option:");
    printf("Usage %s [password]\n",*argv);
    iVar1 = 1;
  }
  else {
    iVar1 = authenticate(argv[1]);
    if (iVar1 == 0) {
      puts("Invalid password. Try again!");
    }
    else {
      system("clear");
      puts("Welcome, you have access to top secret part o
    }
    iVar1 = 0;
  }
  return iVar1;
}
```

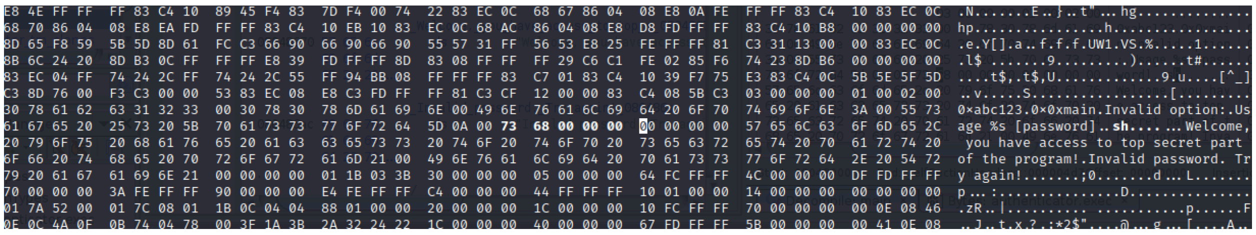


```
                  03 00 00 00 01 00 02 00          ........
78 61 62 63 31 32 33 00 30 78 30 78 6d 61 69    0xabc123.0x0xmai   00110000 01111000 01100001
00 49 6e 76 61 6c 69 64 20 6f 70 74 69 6f 6e    n.Invalid option   01101110 00000000 01001001
00 55 73 61 67 65 20 25 73 20 5b 70 61 73 73    :.Usage %s [pass   00111010 00000000 01010101
6f 72 64 5d 0a 00 73 68 00 00 00 00 00 00 00    word]..sh.......   01110111 01101111 01110010
65 6c 63 6f 6d 65 2c 20 79 6f 75 20 68 61 76    Welcome, you hav   01010111 01100101 01101100
20 61 63 63 65 73 73 20 74 6f 20 74 6f 70 20    e access to top    01100101 00000000 01100001
65 63 72 65 74 20 70 61 72 74 20 6f 66 20 74    secret part of t   01110011 01100101 01100011
65 20 70 72 6f 67 72 61 6d 21 00 49 6e 76 61    he program!.Inva   01101000 01100101 00100000
69 64 20 70 61 73 73 77 6f 72 64 2e 20 54 72    lid password. Tr   01101100 01101001 01100100
20 61 67 61 69 6e 21 00                         y again!.          01111001 00100000 01100001
.. .. .. .. .. .. .. .. .. .. .. .. .. .. ..    ................   ........ ........ ........
```

- I also experimented with hexedit which is capable of doing the same switch of “clear” to “sh”.

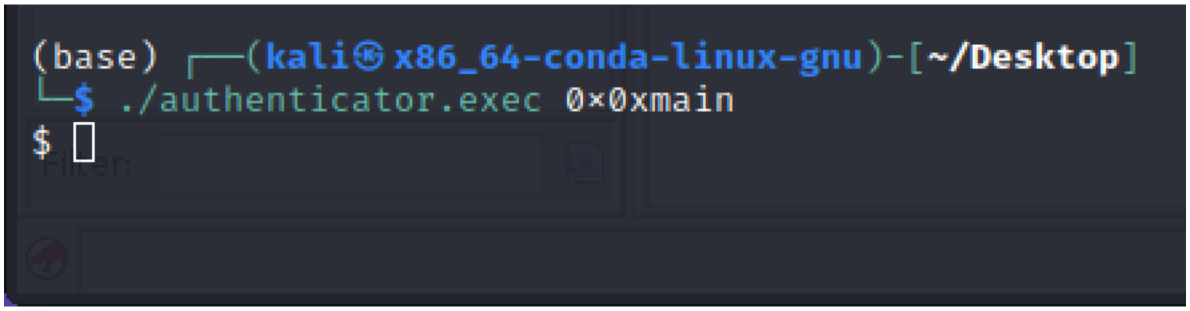


```
E8 4E FF FF  FF 83 C4 10  89 45 F4 83  7D F4 00 74  22 83 EC 0C  68 67 86 04  08 E8 0A FE  FF FF 83 C4  10 83 EC 0C  .N.......E..}..t"...hg.............
68 70 86 04  08 E8 EA FD  FF FF 83 C4  10 EB 10 83  EC 0C 68 AC  86 04 08 E8  D8 FD FF FF  83 C4 10 B8  00 00 00 00  hp................h.................
8D 65 F8 59  5B 5D 8D 61  FC C3 66 90  66 90 66 90  55 57 31 FF  56 53 E8 25  FE FF FF 81  C3 31 13 00  00 83 EC 0C  .e.Y[].a..f.f.f.UW1.VS.%.....1......
8B 6C 24 20  8D B3 0C FF  FF FF E8 39  FD FF FF 8D  83 08 FF FF  FF 29 C6 C1  FE 02 85 F6  74 23 8D B6  00 00 00 00  .l$ .......9.........).....t#......
83 EC 04 FF  74 24 2C FF  74 24 2C 55  FF 94 BB 08  FF FF FF 83  C7 01 83 C4  10 39 F7 75  E3 83 C4 0C  5B 5E 5F 5D  ....t$,.t$,U.............9.u....[^_]
C3 8D 76 00  F3 C3 00 00  53 83 EC 08  E8 C3 FD FF  FF 81 C3 CF  12 00 00 83  C4 08 5B C3  03 00 00 00  01 00 02 00  ..v.....S....................[........
30 78 61 62  63 31 32 33  00 30 78 30  78 6D 61 69  6E 00 49 6E  76 61 6C 69  64 20 6F 70  74 69 6F 6E  3A 00 55 73  0×abc123.0×0xmain.Invalid option:.Us
61 67 65 20  25 73 20 5B  70 61 73 73  77 6F 72 64  5D 0A 00 63  6C 65 61 72  00 00 00 00  57 65 6C 63  6F 6D 65 2C  age %s [password]..clear....Welcome,
20 79 6F 75  20 68 61 76  65 20 61 63  63 65 73 73  20 74 6F 20  74 6F 70 20  73 65 63 72  65 74 20 70  61 72 74 20   you have access to top secret part 
6F 66 20 74  68 65 20 70  72 6F 67 72  61 6D 21 00  49 6E 76 61  6C 69 64 20  70 61 73 73  77 6F 72 64  2E 20 54 72  of the program!.Invalid password. Tr
79 20 61 67  61 69 6E 21  00 00 00 00  01 1B 03 3B  30 00 00 00  05 00 00 00  64 FC FF FF  4C 00 00 00  DF FD FF FF  y again!.......;0.......d...L.......
70 00 00 00  3A FE FF FF  90 00 00 00  E4 FE FF FF  C4 00 00 00  44 FF FF FF  10 01 00 00  14 00 00 00  00 00 00 00  p...:...............D...............
01 7A 52 00  01 7C 08 01  1B 0C 04 04  88 01 00 00  20 00 00 00  1C 00 00 00  10 FC FF FF  70 00 00 00  00 0E 08 46  .zR..|.......... ...........p......F
0E 0C 4A 0F  0B 74 04 78  00 3F 1A 3B  2A 32 24 22  1C 00 00 00  40 00 00 00  67 FD FF FF  5B 00 00 00  00 41 0E 08  ..J..t.x.?.;*2$"....@...g...[....A..
```



```
E8 4E FF FF  FF 83 C4 10  89 45 F4 83  7D F4 00 74  22 83 EC 0C  68 67 86 04  08 E8 0A FE  FF FF 83 C4  10 83 EC 0C  .N.......E..}..t"...hg.............
68 70 86 04  08 E8 EA FD  FF FF 83 C4  10 EB 10 83  EC 0C 68 AC  86 04 08 E8  D8 FD FF FF  83 C4 10 B8  00 00 00 00  hp................h.................
8D 65 F8 59  5B 5D 8D 61  FC C3 66 90  66 90 66 90  55 57 31 FF  56 53 E8 25  FE FF FF 81  C3 31 13 00  00 83 EC 0C  .e.Y[].a..f.f.f.UW1.VS.%.....1......
8B 6C 24 20  8D B3 0C FF  FF FF E8 39  FD FF FF 8D  83 08 FF FF  FF 29 C6 C1  FE 02 85 F6  74 23 8D B6  00 00 00 00  .l$ .......9.........).....t#......
83 EC 04 FF  74 24 2C FF  74 24 2C 55  FF 94 BB 08  FF FF FF 83  C7 01 83 C4  10 39 F7 75  E3 83 C4 0C  5B 5E 5F 5D  ....t$,.t$,U.............9.u....[^_]
C3 8D 76 00  F3 C3 00 00  53 83 EC 08  E8 C3 FD FF  FF 81 C3 CF  12 00 00 83  C4 08 5B C3  03 00 00 00  01 00 02 00  ..v.....S....................[........
30 78 61 62  63 31 32 33  00 30 78 30  78 6D 61 69  6E 00 49 6E  76 61 6C 69  64 20 6F 70  74 69 6F 6E  3A 00 55 73  0×abc123.0×0xmain.Invalid option:.Us
61 67 65 20  25 73 20 5B  70 61 73 73  77 6F 72 64  5D 0A 00 73  68 00 00 00  00 00 00 00  57 65 6C 63  6F 6D 65 2C  age %s [password]..sh.......Welcome,
20 79 6F 75  20 68 61 76  65 20 61 63  63 65 73 73  20 74 6F 20  74 6F 70 20  73 65 63 72  65 74 20 70  61 72 74 20   you have access to top secret part 
6F 66 20 74  68 65 20 70  72 6F 67 72  61 6D 21 00  49 6E 76 61  6C 69 64 20  70 61 73 73  77 6F 72 64  2E 20 54 72  of the program!.Invalid password. Tr
79 20 61 67  61 69 6E 21  00 00 00 00  01 1B 03 3B  30 00 00 00  05 00 00 00  64 FC FF FF  4C 00 00 00  DF FD FF FF  y again!.......;0.......d...L.......
70 00 00 00  3A FE FF FF  90 00 00 00  E4 FE FF FF  C4 00 00 00  44 FF FF FF  10 01 00 00  14 00 00 00  00 00 00 00  p...:...............D...............
01 7A 52 00  01 7C 08 01  1B 0C 04 04  88 01 00 00  20 00 00 00  1C 00 00 00  10 FC FF FF  70 00 00 00  00 0E 08 46  .zR..|.......... ...........p......F
0E 0C 4A 0F  0B 74 04 78  00 3F 1A 3B  2A 32 24 22  1C 00 00 00  40 00 00 00  67 FD FF FF  5B 00 00 00  00 41 0E 08  ..J..t.x.?.;*2$"....@...g...[....A..
```

- Now when I run the authenticator with the correct password it executes the /bin/sh shell program.



```
(base) ┌──(kali㉿x86_64-conda-linux-gnu)-[~/Desktop]
└─$ ./authenticator.exec 0×0xmain
$ 
```