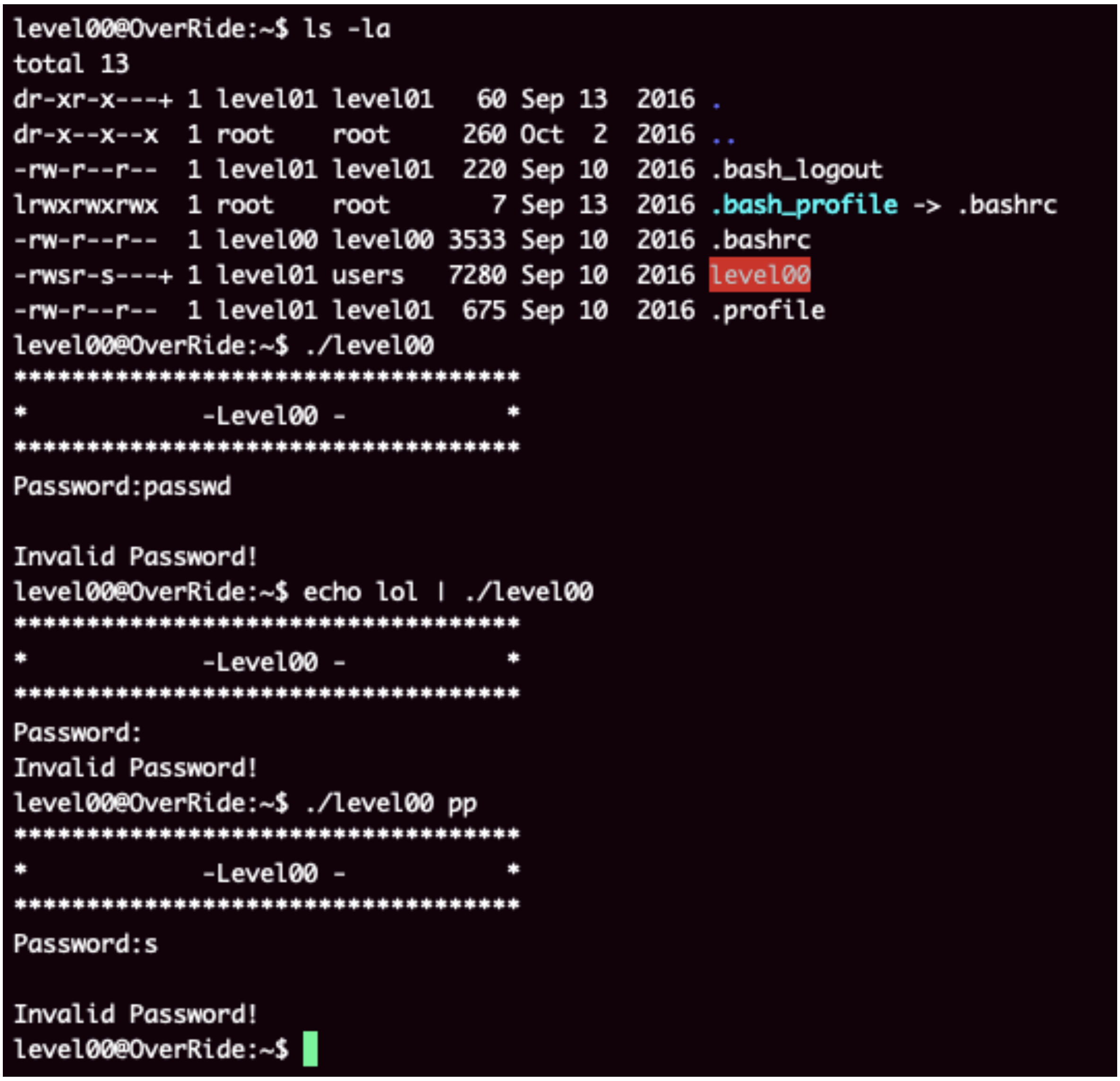
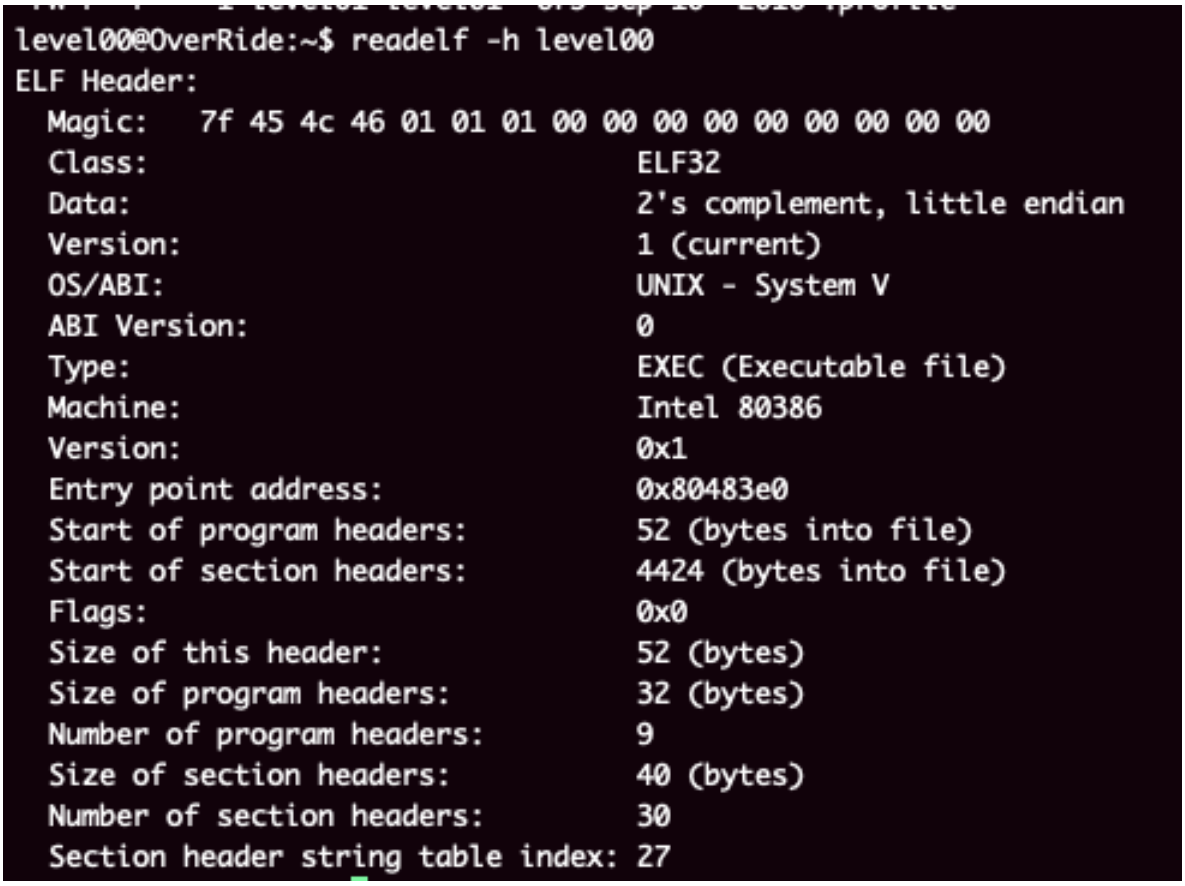
LEVEL00:



```
level00@OverRide:~$ ls -la
total 13
dr-xr-x---+ 1 level01 level01   60 Sep 13  2016 .
dr-x--x--x  1 root    root     260 Oct  2  2016 ..
-rw-r--r--  1 level01 level01  220 Sep 10  2016 .bash_logout
lrwxrwxrwx  1 root    root       7 Sep 13  2016 .bash_profile -> .bashrc
-rw-r--r--  1 level00 level00 3533 Sep 10  2016 .bashrc
-rwsr-s---+ 1 level01 users   7280 Sep 10  2016 level00
-rw-r--r--  1 level01 level01  675 Sep 10  2016 .profile
level00@OverRide:~$ ./level00
***********************************
*            -Level00 -           *
***********************************
Password:passwd

Invalid Password!
level00@OverRide:~$ echo lol | ./level00
***********************************
*            -Level00 -           *
***********************************
Password:
Invalid Password!
level00@OverRide:~$ ./level00 pp
***********************************
*            -Level00 -           *
***********************************
Password:s

Invalid Password!
level00@OverRide:~$ 
```

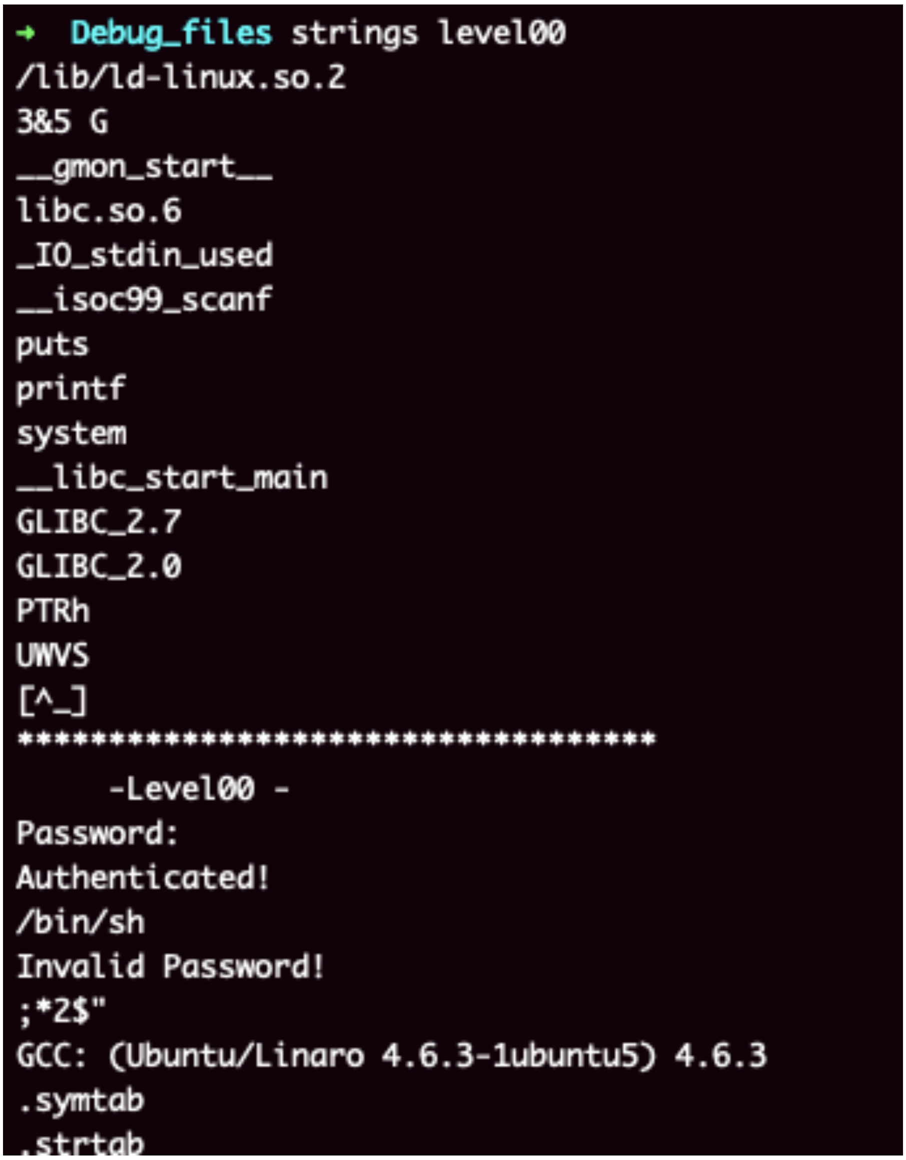
The program is reading on stdin to get a password.

- Readelf -h:



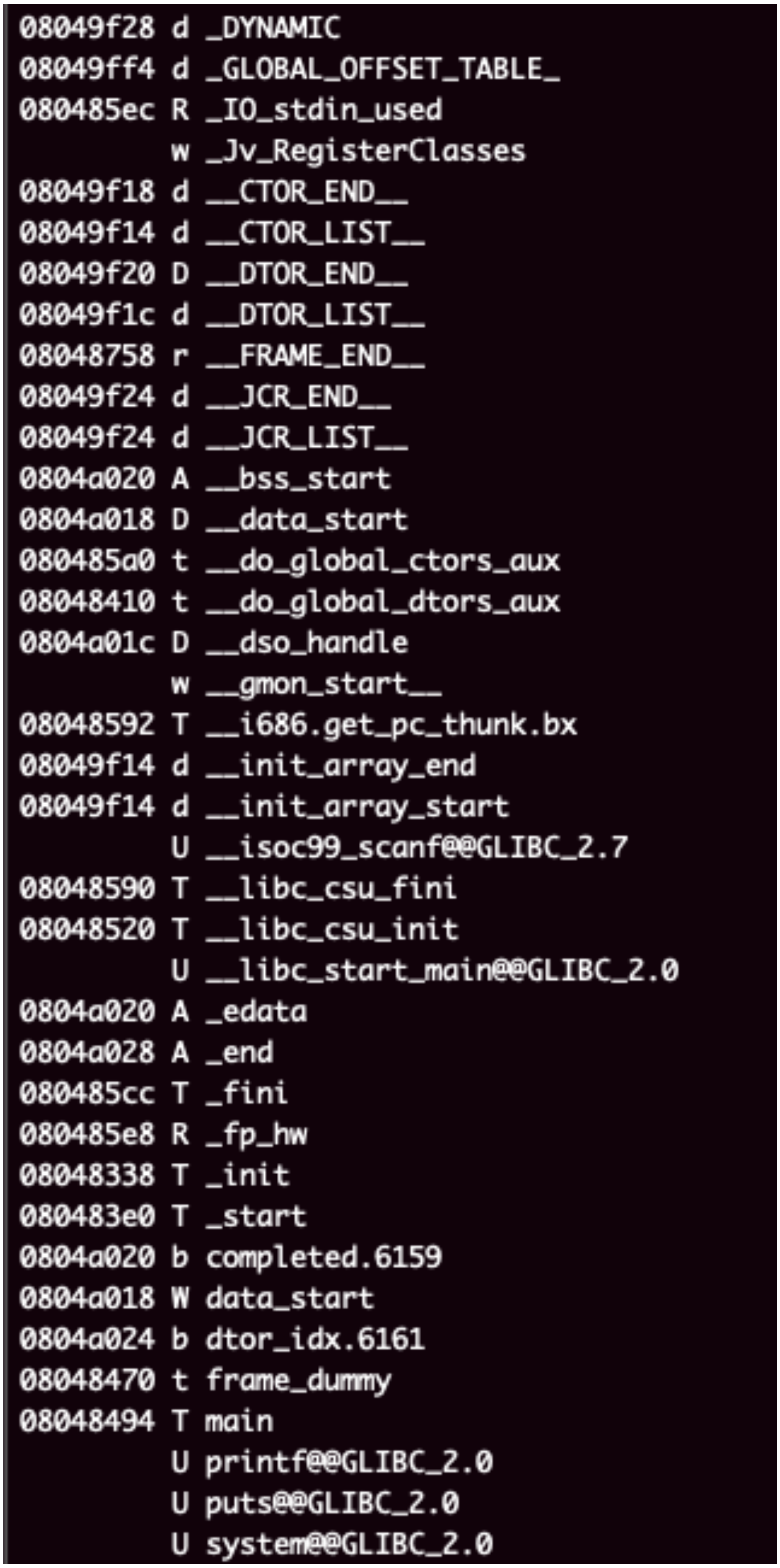
```
level00@OverRide:~$ readelf -h level00
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x80483e0
  Start of program headers:          52 (bytes into file)
  Start of section headers:          4424 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         9
  Size of section headers:           40 (bytes)
  Number of section headers:         30
  Section header string table index: 27
```

- Strings:



```
➜  Debug_files strings level00
/lib/ld-linux.so.2
3&5 G
__gmon_start__
libc.so.6
_IO_stdin_used
__isoc99_scanf
puts
printf
system
__libc_start_main
GLIBC_2.7
GLIBC_2.0
PTRh
UWVS
[^_]
***********************************
     -Level00 -
Password:
Authenticated!
/bin/sh
Invalid Password!
;*2$"
GCC: (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
.symtab
.strtab
```

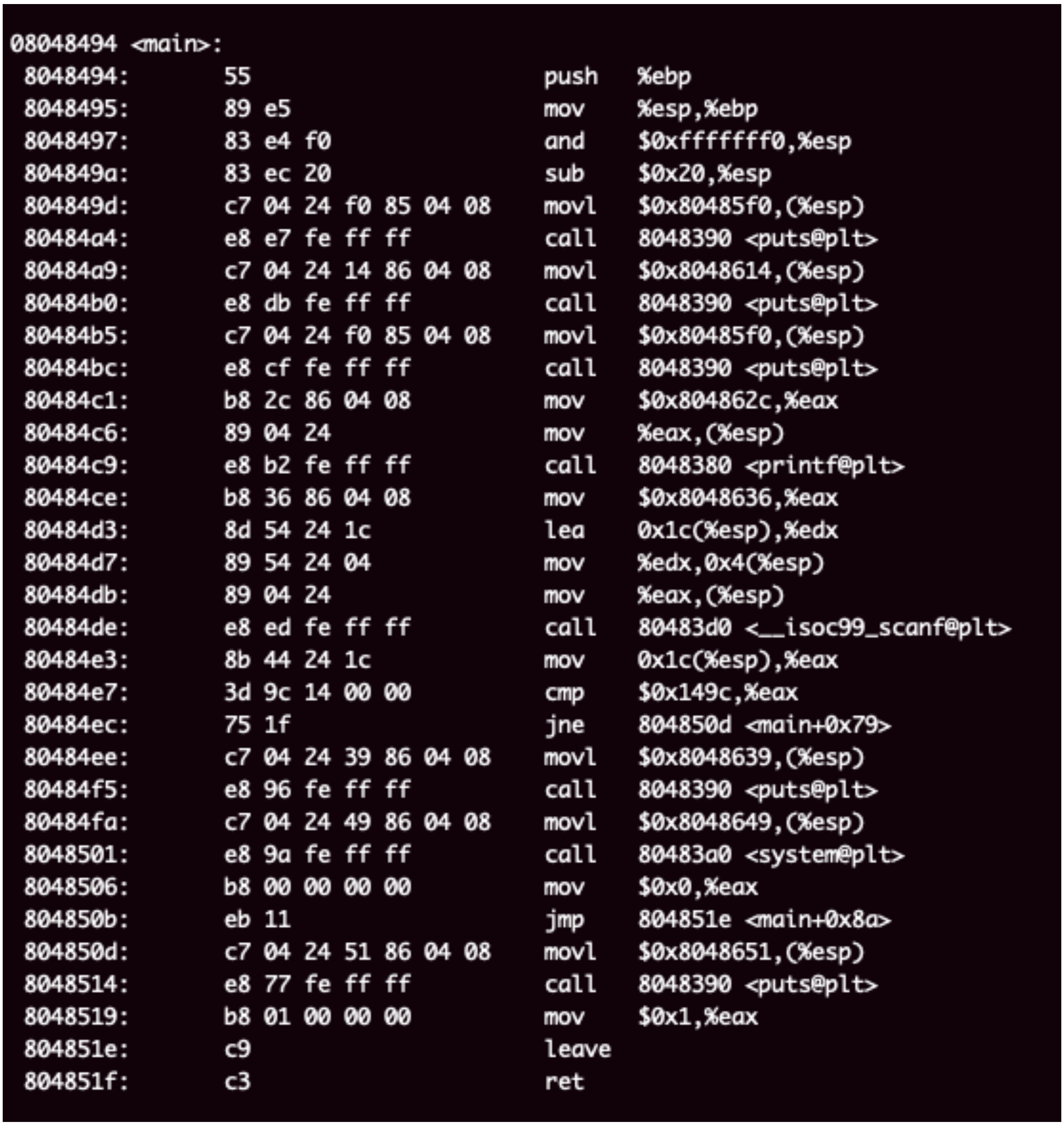
- Nm:



```
08049f28 d _DYNAMIC
08049ff4 d _GLOBAL_OFFSET_TABLE_
080485ec R _IO_stdin_used
         w _Jv_RegisterClasses
08049f18 d __CTOR_END__
08049f14 d __CTOR_LIST__
08049f20 D __DTOR_END__
08049f1c d __DTOR_LIST__
08048758 r __FRAME_END__
08049f24 d __JCR_END__
08049f24 d __JCR_LIST__
0804a020 A __bss_start
0804a018 D __data_start
080485a0 t __do_global_ctors_aux
08048410 t __do_global_dtors_aux
0804a01c D __dso_handle
         w __gmon_start__
08048592 T __i686.get_pc_thunk.bx
08049f14 d __init_array_end
08049f14 d __init_array_start
         U __isoc99_scanf@@GLIBC_2.7
08048590 T __libc_csu_fini
08048520 T __libc_csu_init
         U __libc_start_main@@GLIBC_2.0
0804a020 A _edata
0804a028 A _end
080485cc T _fini
080485e8 R _fp_hw
08048338 T _init
080483e0 T _start
0804a020 b completed.6159
0804a018 W data_start
0804a024 b dtor_idx.6161
08048470 t frame_dummy
08048494 T main
         U printf@@GLIBC_2.0
         U puts@@GLIBC_2.0
         U system@@GLIBC_2.0
```

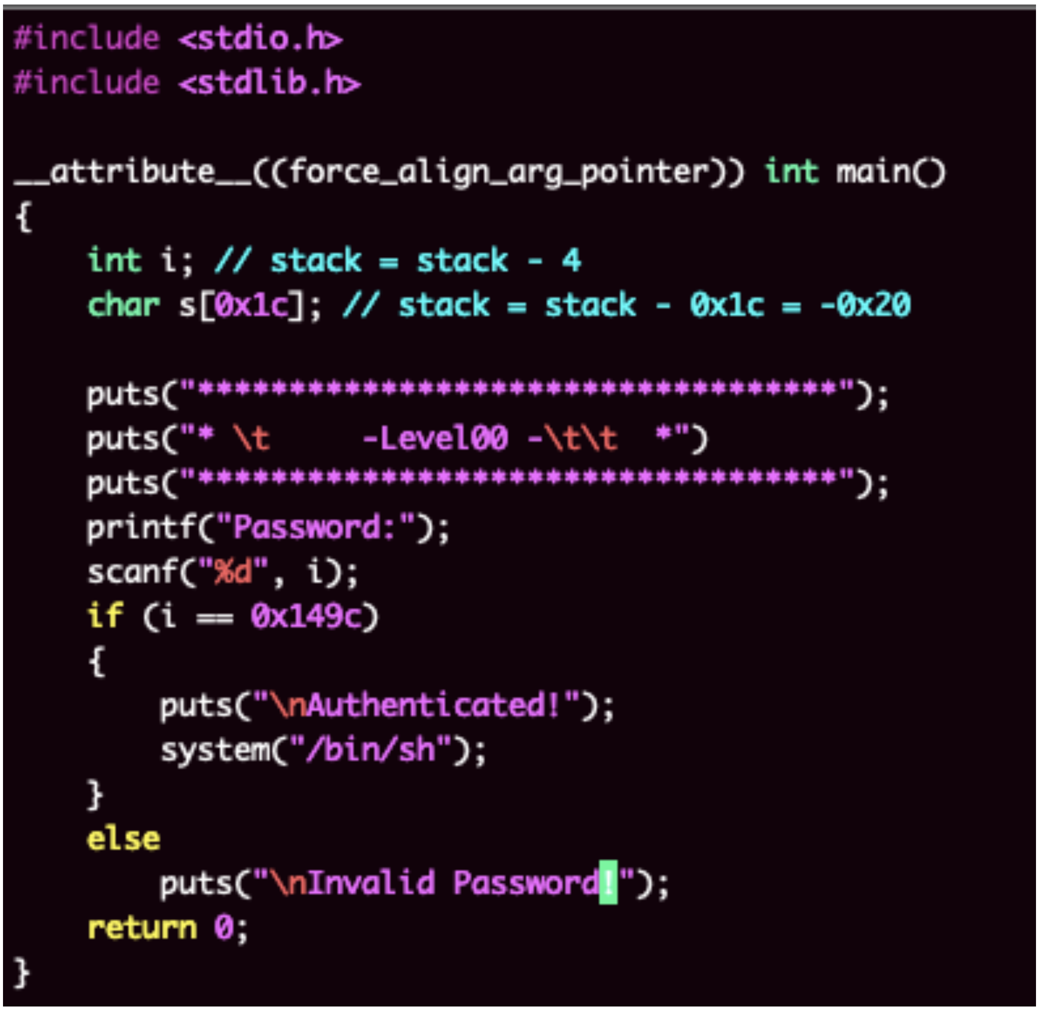
We are on a 32bit compiled binary.

- Objdump -d:



```
08048494 <main>:
 8048494:       55                      push   %ebp
 8048495:       89 e5                   mov    %esp,%ebp
 8048497:       83 e4 f0                and    $0xfffffff0,%esp
 804849a:       83 ec 20                sub    $0x20,%esp
 804849d:       c7 04 24 f0 85 04 08    movl   $0x80485f0,(%esp)
 80484a4:       e8 e7 fe ff ff          call   8048390 <puts@plt>
 80484a9:       c7 04 24 14 86 04 08    movl   $0x8048614,(%esp)
 80484b0:       e8 db fe ff ff          call   8048390 <puts@plt>
 80484b5:       c7 04 24 f0 85 04 08    movl   $0x80485f0,(%esp)
 80484bc:       e8 cf fe ff ff          call   8048390 <puts@plt>
 80484c1:       b8 2c 86 04 08          mov    $0x804862c,%eax
 80484c6:       89 04 24                mov    %eax,(%esp)
 80484c9:       e8 b2 fe ff ff          call   8048380 <printf@plt>
 80484ce:       b8 36 86 04 08          mov    $0x8048636,%eax
 80484d3:       8d 54 24 1c             lea    0x1c(%esp),%edx
 80484d7:       89 54 24 04             mov    %edx,0x4(%esp)
 80484db:       89 04 24                mov    %eax,(%esp)
 80484de:       e8 ed fe ff ff          call   80483d0 <__isoc99_scanf@plt>
 80484e3:       8b 44 24 1c             mov    0x1c(%esp),%eax
 80484e7:       3d 9c 14 00 00          cmp    $0x149c,%eax
 80484ec:       75 1f                   jne    804850d <main+0x79>
 80484ee:       c7 04 24 39 86 04 08    movl   $0x8048639,(%esp)
 80484f5:       e8 96 fe ff ff          call   8048390 <puts@plt>
 80484fa:       c7 04 24 49 86 04 08    movl   $0x8048649,(%esp)
 8048501:       e8 9a fe ff ff          call   80483a0 <system@plt>
 8048506:       b8 00 00 00 00          mov    $0x0,%eax
 804850b:       eb 11                   jmp    804851e <main+0x8a>
 804850d:       c7 04 24 51 86 04 08    movl   $0x8048651,(%esp)
 8048514:       e8 77 fe ff ff          call   8048390 <puts@plt>
 8048519:       b8 01 00 00 00          mov    $0x1,%eax
 804851e:       c9                      leave
 804851f:       c3                      ret
```

Source:

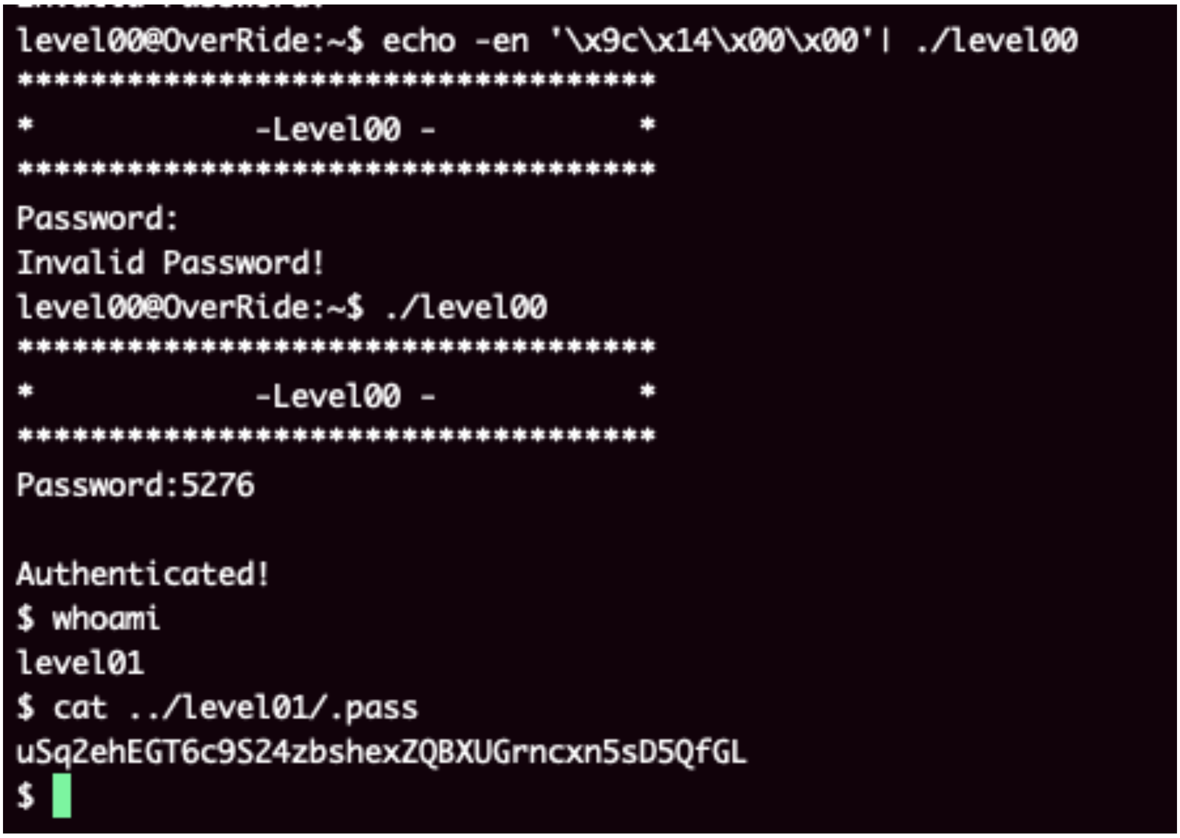


```c
#include <stdio.h>
#include <stdlib.h>

__attribute__((force_align_arg_pointer)) int main()
{
    int i; // stack = stack - 4
    char s[0x1c]; // stack = stack - 0x1c = -0x20

    puts("***********************************");
    puts("* \t      -Level00 -\t\t  *")
    puts("***********************************");
    printf("Password:");
    scanf("%d", i);
    if (i == 0x149c)
    {
        puts("\nAuthenticated!");
        system("/bin/sh");
    }
    else
        puts("\nInvalid Password!");
    return 0;
}
```

Our input is formatted by scanf from characters to decimal. Like atoi(). Not from the value of the octet (0x14 et 0x9c), but from the value of the formatting from the input characters.

What gives 0x149c in decimal is 5276. So atoi(« 5276 ») return a decimal number 5276, in hexa 0x149c

So the password is « 5276 ».



```
level00@OverRide:~$ echo -en '\x9c\x14\x00\x00'| ./level00
***********************************
*            -Level00 -           *
***********************************
Password:
Invalid Password!
level00@OverRide:~$ ./level00
***********************************
*            -Level00 -           *
***********************************
Password:5276

Authenticated!
$ whoami
level01
$ cat ../level01/.pass
uSq2ehEGT6c9S24zbshexZQBXUGrncxn5sD5QfGL
$ 
```

Flag : uSq2ehEGT6c9S24zbshexZQBXUGrncxn5sD5QfGL