Level1 by Sudo0x18

Preface

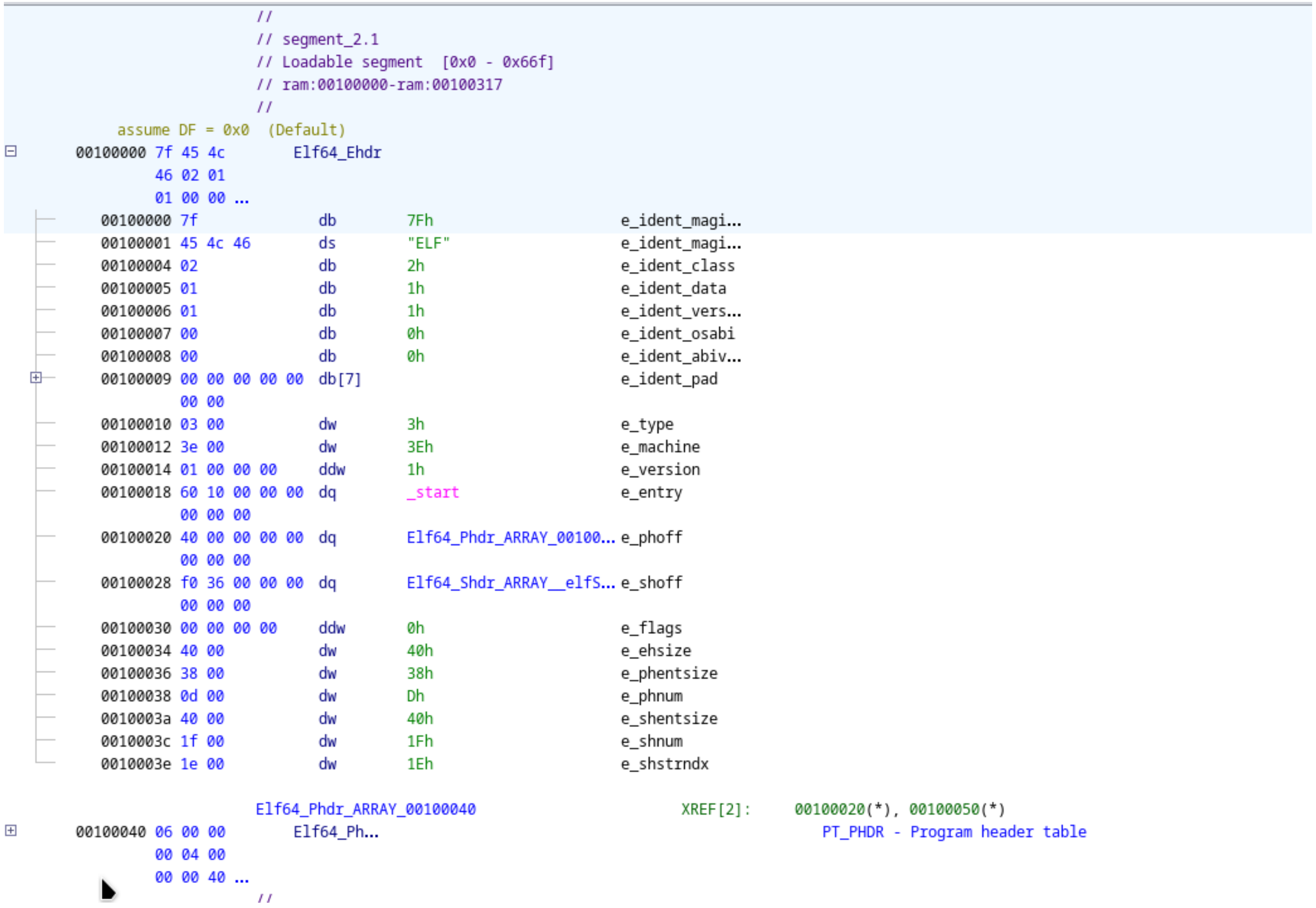
The first Unix CrackMe on crackmes.one, the entry is crackable and can be finished in 5 minutes or less, even for beginners.

Required Tools:

- Ghidra

Recon

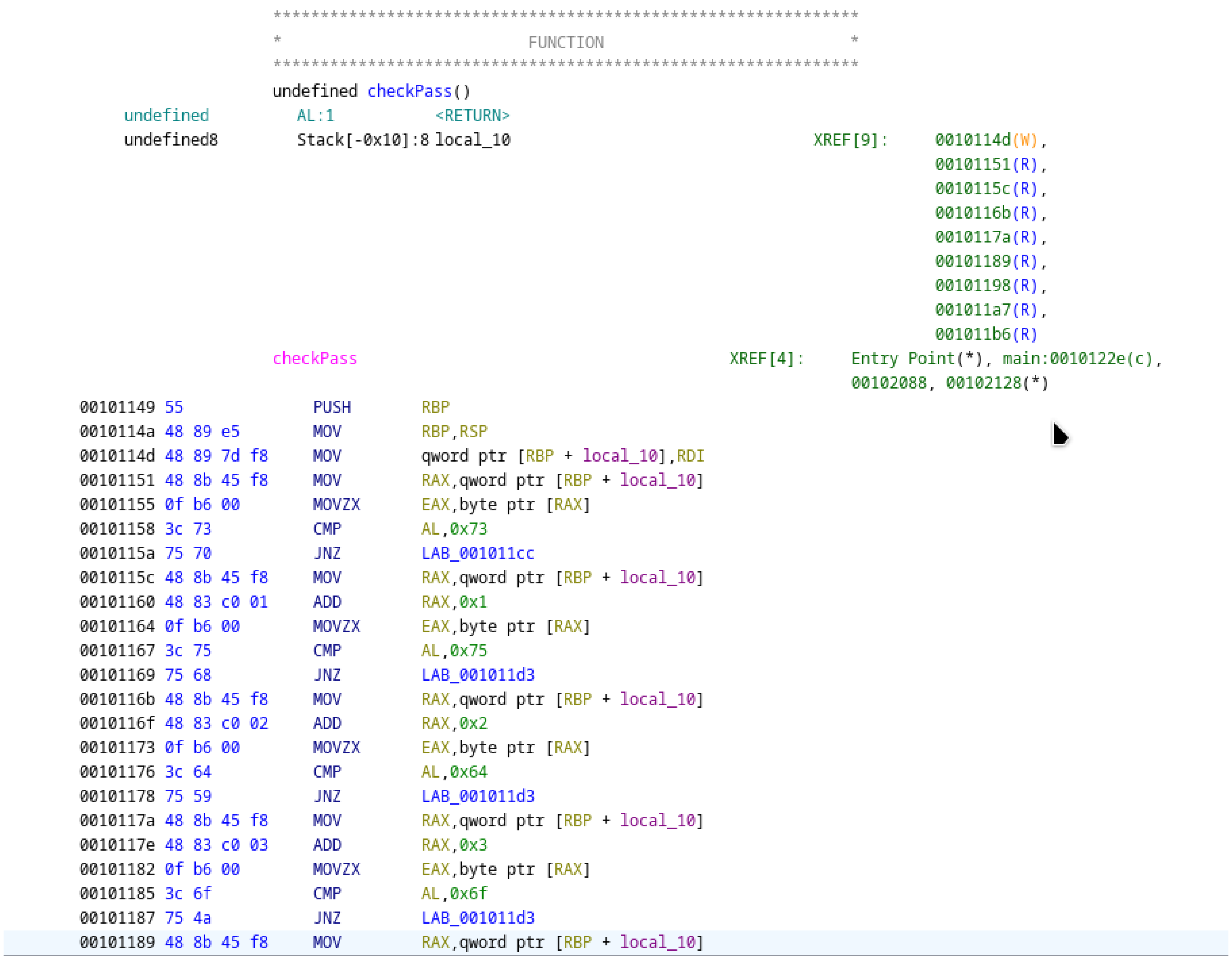
Using Ghidra's Code browser, extracting the `level1` file into a folder and placing the file in Ghidra the decompiled file in assembly is generated.



```
                             //
                             // segment_2.1
                             // Loadable segment  [0x0 - 0x66f]
                             // ram:00100000-ram:00100317
                             //
                   assume DF = 0x0  (Default)
        00100000 7f 45 4c        Elf64_Ehdr
                 46 02 01
                 01 00 00 ...
           00100000 7f           db        7Fh                e_ident_magi…
           00100001 45 4c 46     ds        "ELF"              e_ident_magi…
           00100004 02           db        2h                 e_ident_class
           00100005 01           db        1h                 e_ident_data
           00100006 01           db        1h                 e_ident_vers…
           00100007 00           db        0h                 e_ident_osabi
           00100008 00           db        0h                 e_ident_abiv…
           00100009 00 00 00 00 00 db[7]                      e_ident_pad
                    00 00
           00100010 03 00        dw        3h                 e_type
           00100012 3e 00        dw        3Eh                e_machine
           00100014 01 00 00 00  ddw       1h                 e_version
           00100018 60 10 00 00 00 dq      _start             e_entry
                    00 00 00
           00100020 40 00 00 00 00 dq      Elf64_Phdr_ARRAY_00100… e_phoff
                    00 00 00
           00100028 f0 36 00 00 00 dq      Elf64_Shdr_ARRAY__elfS… e_shoff
                    00 00 00
           00100030 00 00 00 00  ddw       0h                 e_flags
           00100034 40 00        dw        40h                e_ehsize
           00100036 38 00        dw        38h                e_phentsize
           00100038 0d 00        dw        Dh                 e_phnum
           0010003a 40 00        dw        40h                e_shentsize
           0010003c 1f 00        dw        1Fh                e_shnum
           0010003e 1e 00        dw        1Eh                e_shstrndx

                             Elf64_Phdr_ARRAY_00100040              XREF[2]:     00100020(*), 00100050(*)
        00100040 06 00 00        Elf64_Ph…                                          PT_PHDR - Program header table
                 00 04 00
                 00 00 40 ...
                             //
```

First, we attempt to find an undefined function. The first given undefined function is `checkPass()`. Through inference, it can be suggested that this function checks the input code's validity.



```
                    **************************************************************
                    *                          FUNCTION                          *
                    **************************************************************
                    undefined checkPass()
     undefined        AL:1           <RETURN>
     undefined8       Stack[-0x10]:8 local_10                       XREF[9]:     0010114d(W),
                                                                                 00101151(R),
                                                                                 0010115c(R),
                                                                                 0010116b(R),
                                                                                 0010117a(R),
                                                                                 00101189(R),
                                                                                 00101198(R),
                                                                                 001011a7(R),
                                                                                 001011b6(R)
                    checkPass                                   XREF[4]:     Entry Point(*), main:0010122e(c),
                                                                             00102088, 00102128(*)
00101149 55              PUSH       RBP
0010114a 48 89 e5        MOV        RBP,RSP
0010114d 48 89 7d f8     MOV        qword ptr [RBP + local_10],RDI
00101151 48 8b 45 f8     MOV        RAX,qword ptr [RBP + local_10]
00101155 0f b6 00        MOVZX      EAX,byte ptr [RAX]
00101158 3c 73           CMP        AL,0x73
0010115a 75 70           JNZ        LAB_001011cc
0010115c 48 8b 45 f8     MOV        RAX,qword ptr [RBP + local_10]
00101160 48 83 c0 01     ADD        RAX,0x1
00101164 0f b6 00        MOVZX      EAX,byte ptr [RAX]
00101167 3c 75           CMP        AL,0x75
00101169 75 68           JNZ        LAB_001011d3
0010116b 48 8b 45 f8     MOV        RAX,qword ptr [RBP + local_10]
0010116f 48 83 c0 02     ADD        RAX,0x2
00101173 0f b6 00        MOVZX      EAX,byte ptr [RAX]
00101176 3c 64           CMP        AL,0x64
00101178 75 59           JNZ        LAB_001011d3
0010117a 48 8b 45 f8     MOV        RAX,qword ptr [RBP + local_10]
0010117e 48 83 c0 03     ADD        RAX,0x3
00101182 0f b6 00        MOVZX      EAX,byte ptr [RAX]
00101185 3c 6f           CMP        AL,0x6f
00101187 75 4a           JNZ        LAB_001011d3
00101189 48 8b 45 f8     MOV        RAX,qword ptr [RBP + local_10]
```

By using CheckPass (Ctrl + E), a C pseudocode listing of the function is generated.

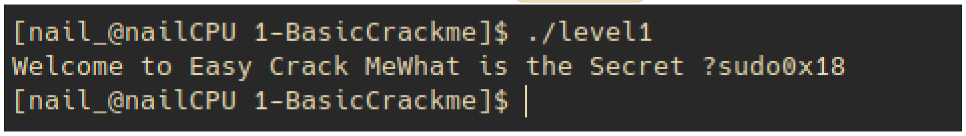
```c
char checkPass(char *param_1)

{
  char cVar1;

  if (*param_1 == 's') {
    cVar1 = param_1[1];
    if ((((cVar1 == 'u') && (cVar1 = param_1[2], cVar1 == 'd')) &&
        (cVar1 = param_1[3], cVar1 == 'o')) &&
       (((cVar1 = param_1[4], cVar1 == '0' && (cVar1 = param_1[5], cVar1 == 'x')) &&
        ((cVar1 = param_1[6], cVar1 == '1' && (cVar1 = param_1[7], cVar1 == '8')))))) {
      cVar1 = '\x01';
    }
  }
  else {
    cVar1 = '\0';
  }
  return cVar1;
}
```

The code checks is the given input with char data type, `param_1` contains is first equal to "s" , which then passes the input into `cVar1`'s 1st index if true. A nested if statement checks if `cVar1` is equal to "u" and the 2nd index is equal to "d". The nested if statement continuously checks if the succeeding input follows the string `sudo0x18`.

We can get the resulting code input as `sudo0x18`, the submitter's username.



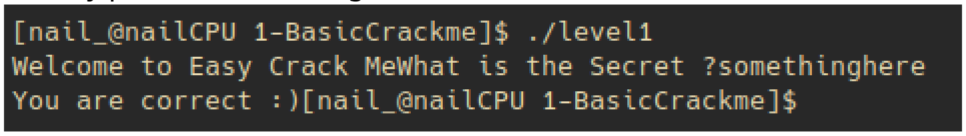
```
[nail_@nailCPU 1-BasicCrackme]$ ./level1
Welcome to Easy Crack MeWhat is the Secret ?sudo0x18
[nail_@nailCPU 1-BasicCrackme]$ |
```

Further Analysis

The C code contains logical errors. The first if statement:

```
if (*param_1 == 's') {
    cVar1 = param_1[1];
```

creates an error where every succeeding input does not affect the results as input into `cVar1` is already passed and unchaged.



```
[nail_@nailCPU 1-BasicCrackme]$ ./level1
Welcome to Easy Crack MeWhat is the Secret ?somethinghere
You are correct :)[nail_@nailCPU 1-BasicCrackme]$
```