PE & ELF parser fuzzing

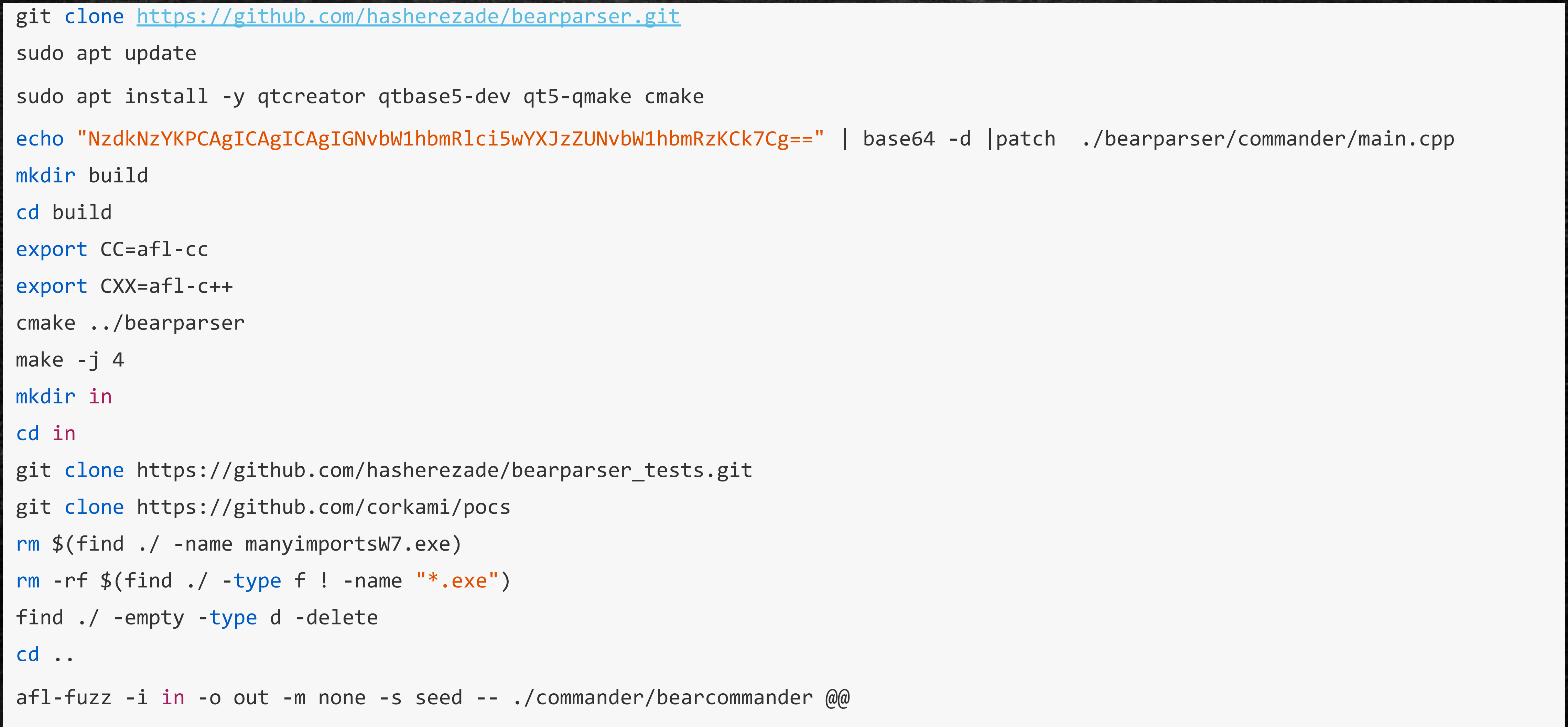
第四組
310555002 高瑋哲、310551052 徐曼妮、310553037 王欣楷

bearparser

Introduction

- Target: bearparser (commit f99ddb8)
- fuzzing tool: afl++

Fuzzing Script



```
git clone https://github.com/hasherezade/bearparser.git
sudo apt update
sudo apt install -y qtcreator qtbase5-dev qt5-qmake cmake
echo "NzdkNzYKPCAgICAgICAgIGNvbW1hbmRlci5wYXJzZUNvbW1hbmRzKCk7Cg==" | base64 -d |patch  ./bearparser/commander/main.cpp
mkdir build
cd build
export CC=afl-cc
export CXX=afl-c++
cmake ../bearparser
make -j 4
mkdir in
cd in
git clone https://github.com/hasherezade/bearparser_tests.git
git clone https://github.com/corkami/pocs
rm $(find ./ -name manyimportsW7.exe)
rm -rf $(find ./ -type f ! -name "*.exe")
find ./ -empty -type d -delete
cd ..
afl-fuzz -i in -o out -m none -s seed -- ./commander/bearcommander @@
```

What we found

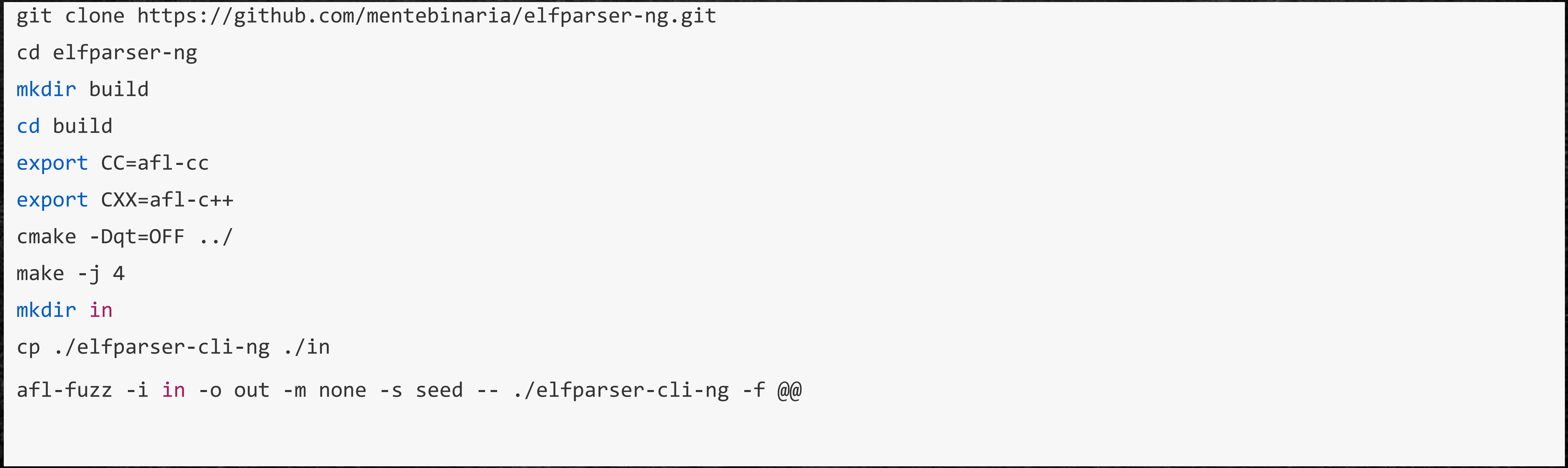
- heap-use-after-free
- https://github.com/hasherezade/bearparser/issues/14

elfparser-ng

Introduction

- Target: elfparser-ng (commit c0bbb5d)
  - An maintained fork of the great ELF Parser.
- fuzzing tool: afl++

Fuzzing Script

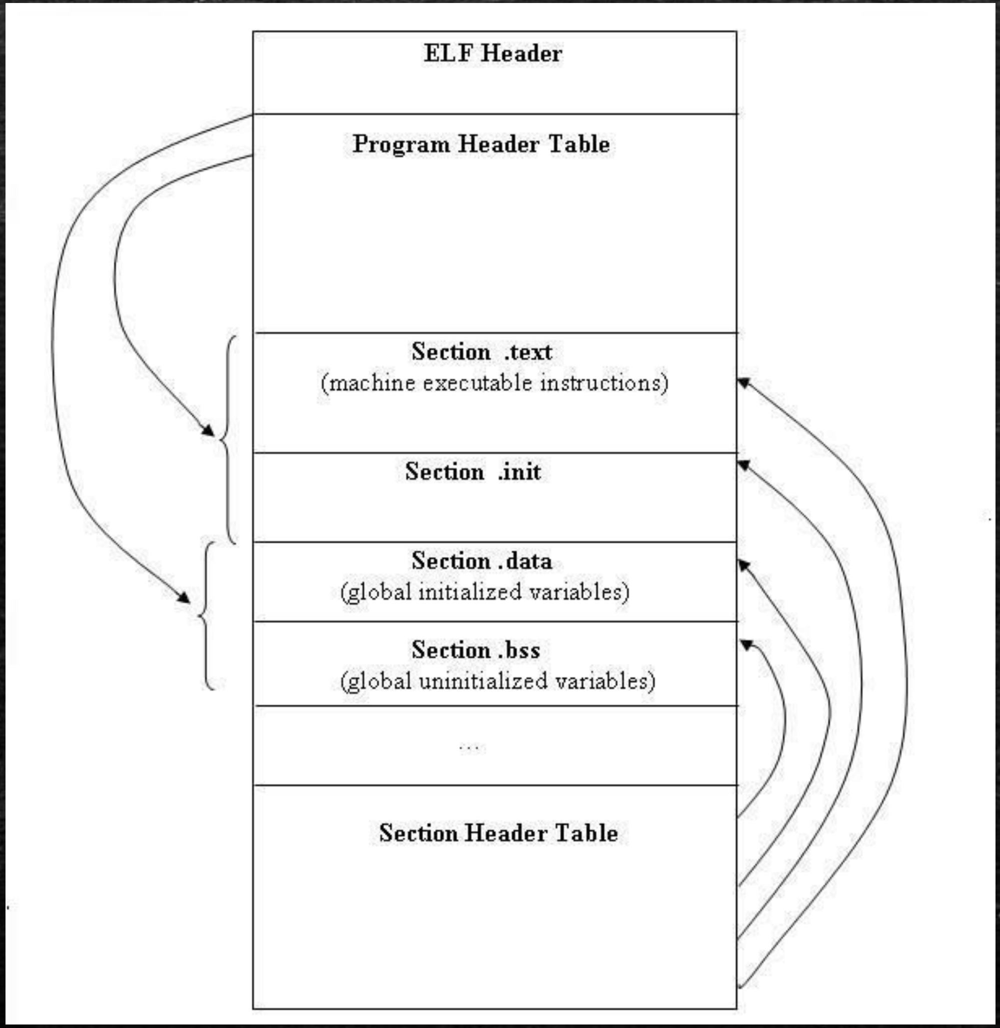


```
git clone https://github.com/mentebinaria/elfparser-ng.git
cd elfparser-ng
mkdir build
cd build
export CC=afl-cc
export CXX=afl-c++
cmake -Dqt=OFF ../
make -j 4
mkdir in
cp ./elfparser-cli-ng ./in
afl-fuzz -i in -o out -m none -s seed -- ./elfparser-cli-ng -f @@
```

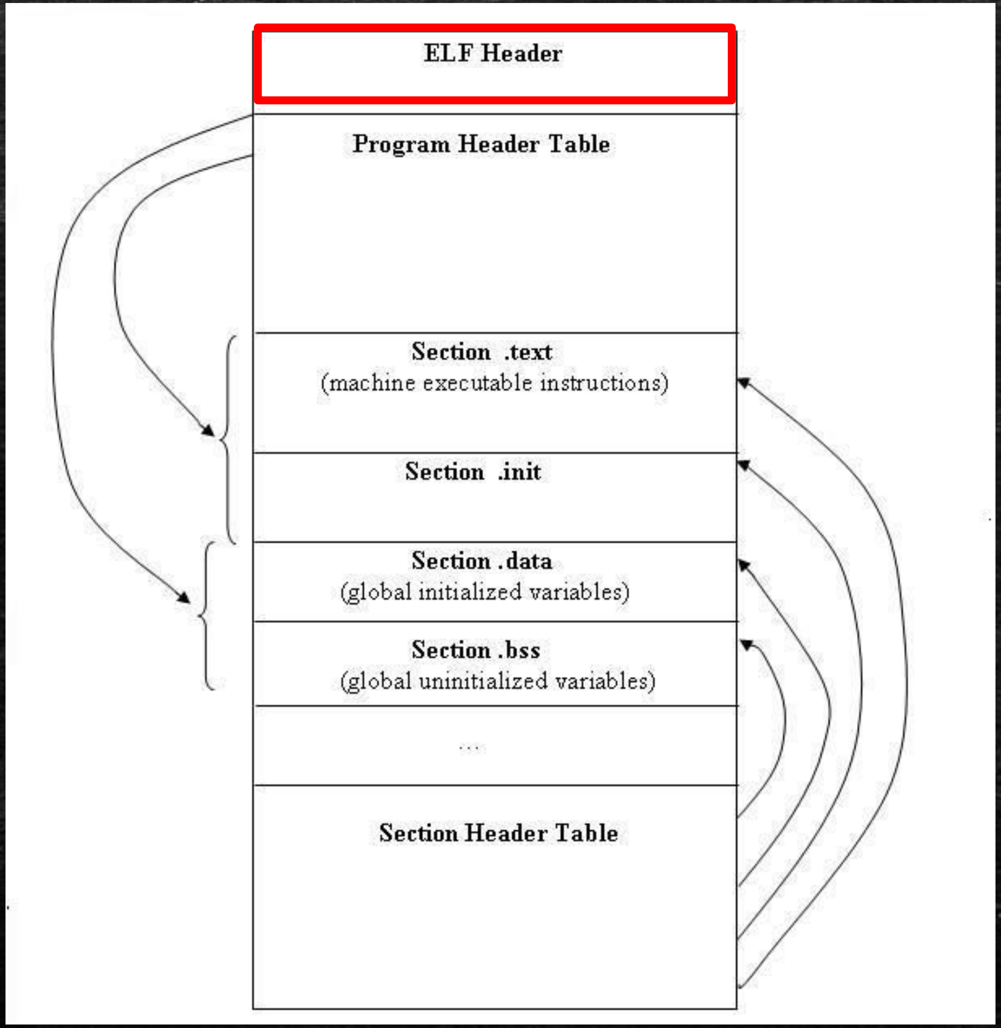
What we found

- SEGV on unknown address
- https://github.com/mentebinaria/elfparser-ng/issues/7

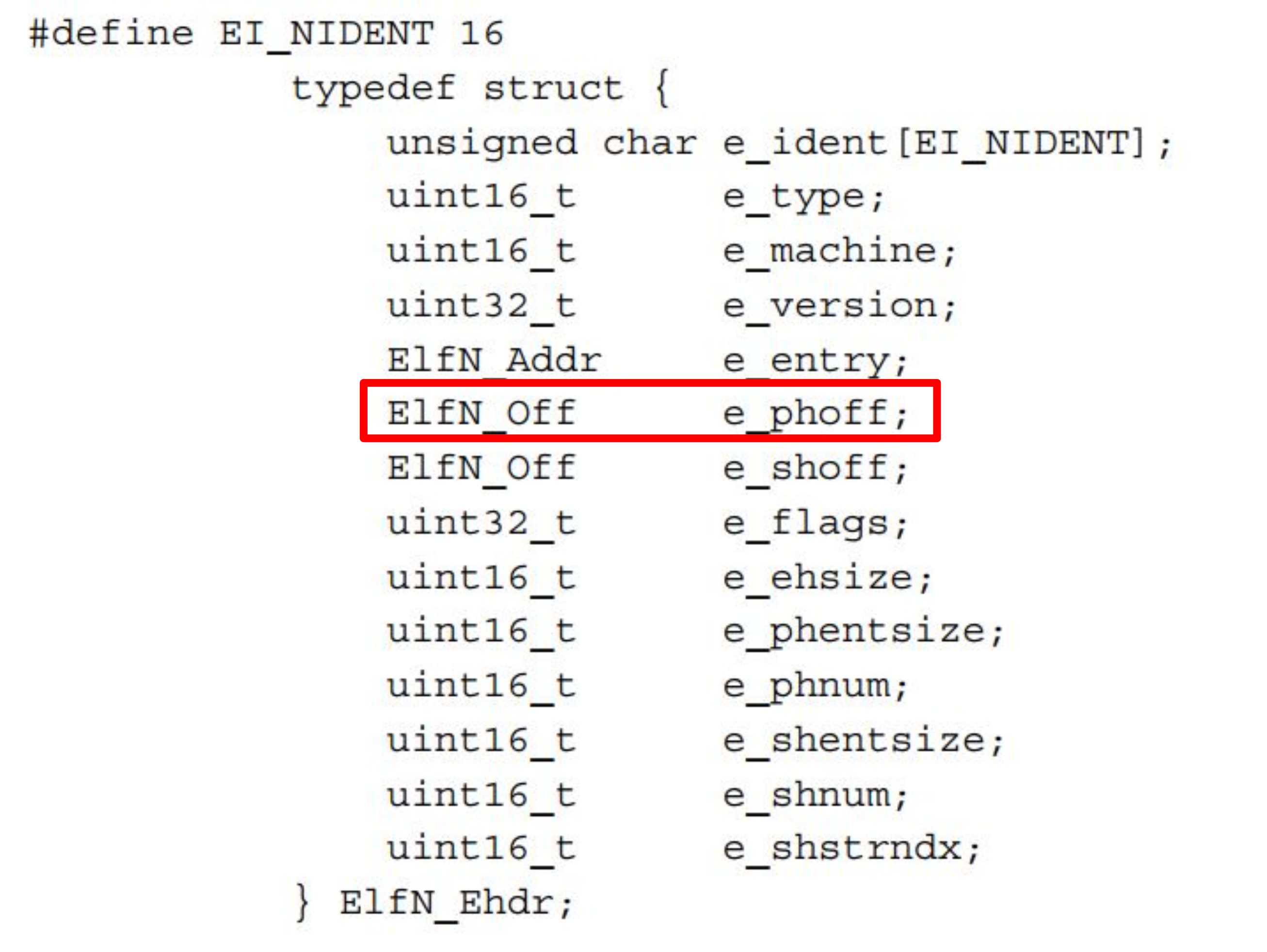
ELF executable format



ELF executable format

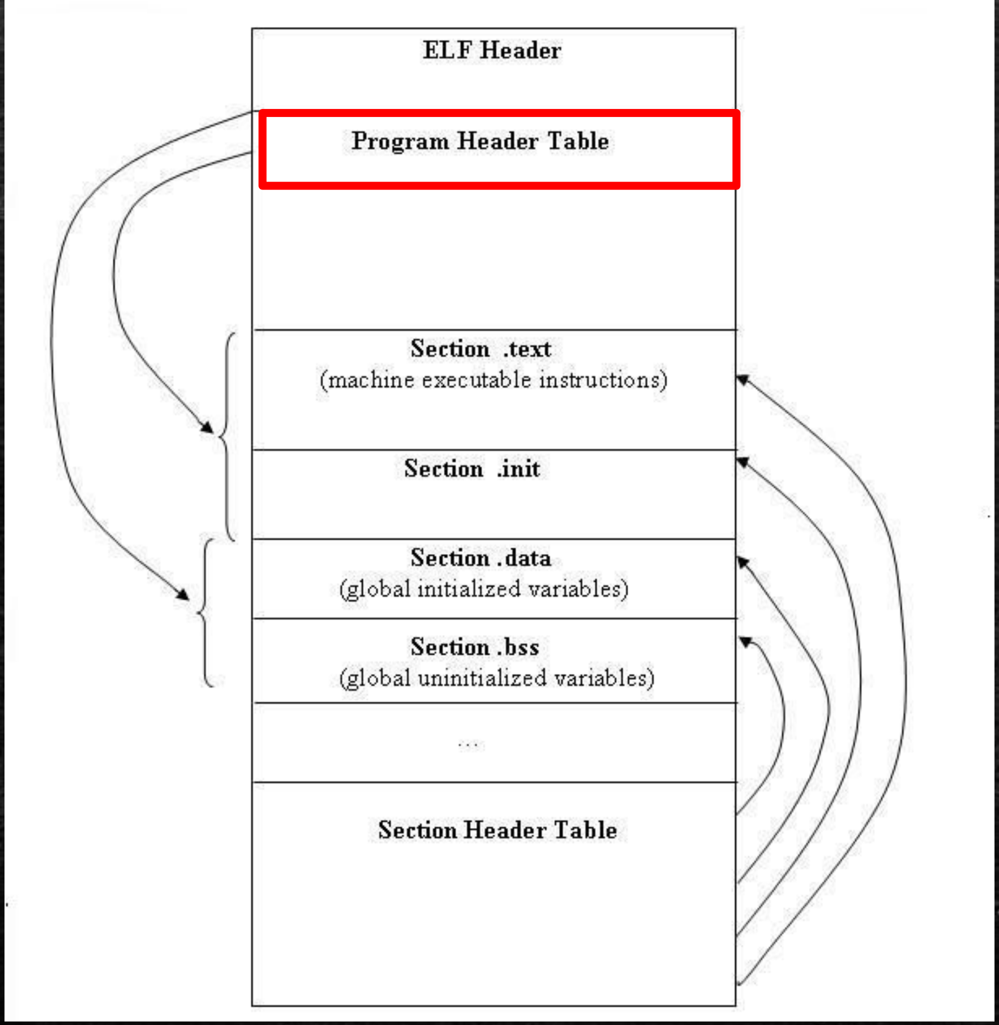


ELF executable format



```
#define EI_NIDENT 16
           typedef struct {
               unsigned char e_ident[EI_NIDENT];
               uint16_t      e_type;
               uint16_t      e_machine;
               uint32_t      e_version;
               ElfN_Addr     e_entry;
               ElfN_Off      e_phoff;
               ElfN_Off      e_shoff;
               uint32_t      e_flags;
               uint16_t      e_ehsize;
               uint16_t      e_phentsize;
               uint16_t      e_phnum;
               uint16_t      e_shentsize;
               uint16_t      e_shnum;
               uint16_t      e_shstrndx;
           } ElfN_Ehdr;
```

ELF executable format



ELF Header

Program Header Table

Section .text
(machine executable instructions)

Section .init

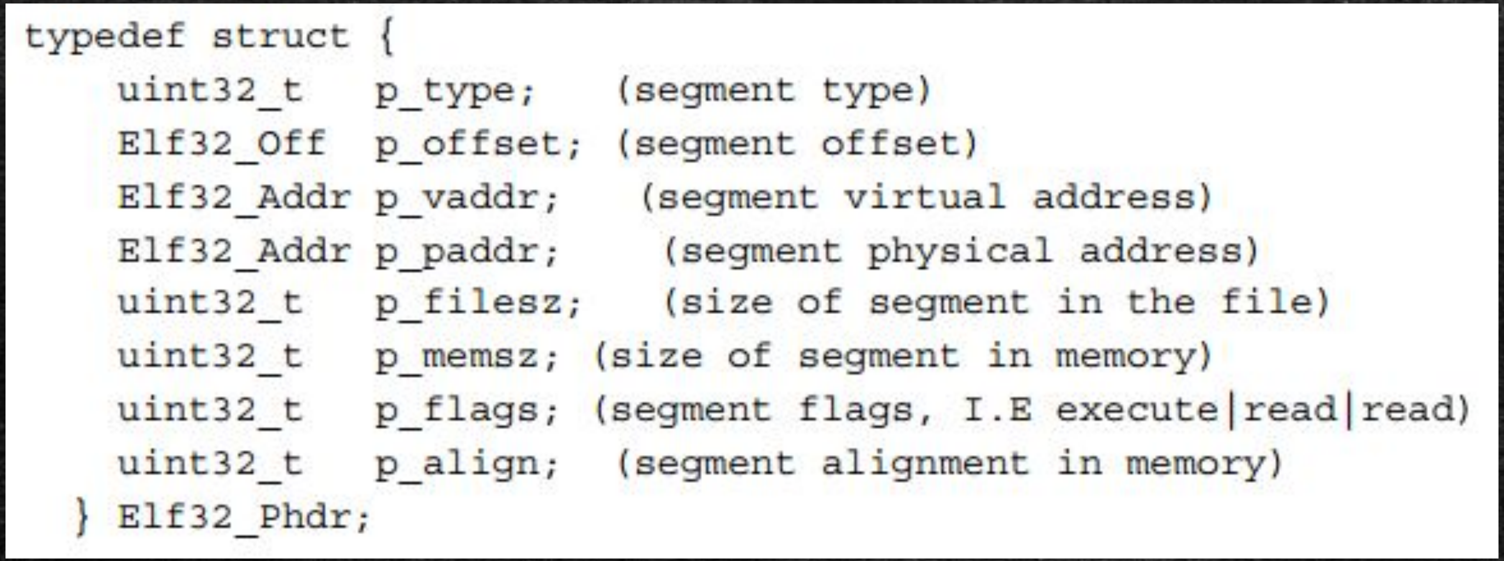
Section .data
(global initialized variables)

Section .bss
(global uninitialized variables)

...

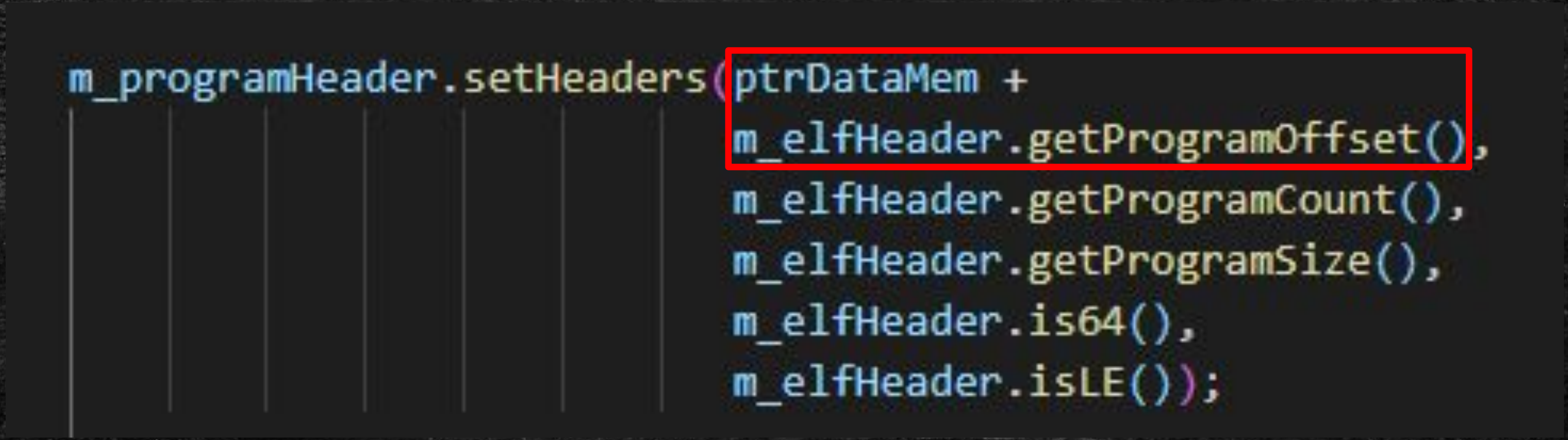
Section Header Table

ELF executable format



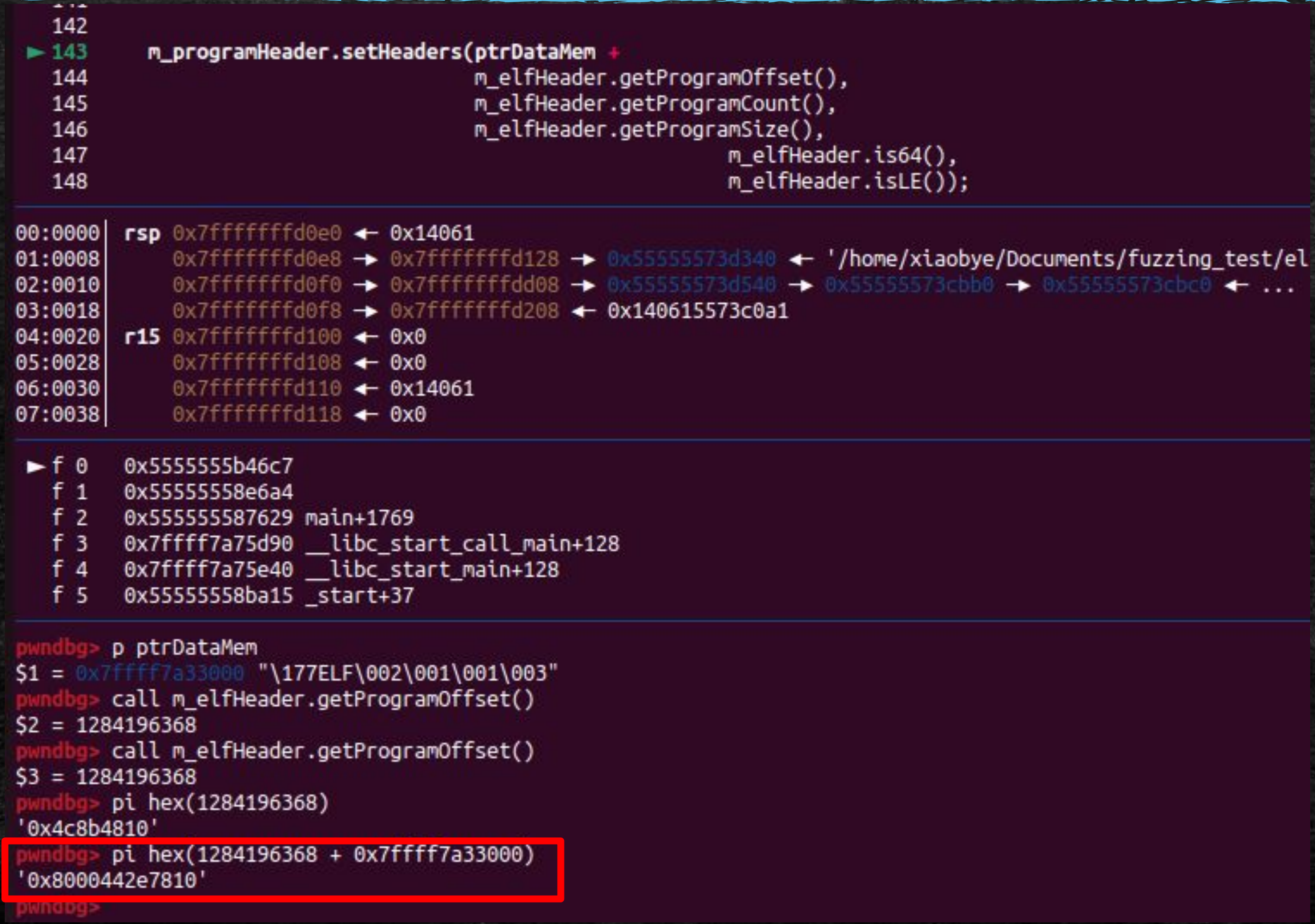
```
typedef struct {
    uint32_t   p_type;   (segment type)
    Elf32_Off  p_offset; (segment offset)
    Elf32_Addr p_vaddr;   (segment virtual address)
    Elf32_Addr p_paddr;    (segment physical address)
    uint32_t   p_filesz;   (size of segment in the file)
    uint32_t   p_memsz; (size of segment in memory)
    uint32_t   p_flags; (segment flags, I.E execute|read|read)
    uint32_t   p_align;  (segment alignment in memory)
  } Elf32_Phdr;
```

Details about bugs



```
m_programHeader.setHeaders(ptrDataMem +
                           m_elfHeader.getProgramOffset(),
                           m_elfHeader.getProgramCount(),
                           m_elfHeader.getProgramSize(),
                           m_elfHeader.is64(),
                           m_elfHeader.isLE());
```

Details about bugs



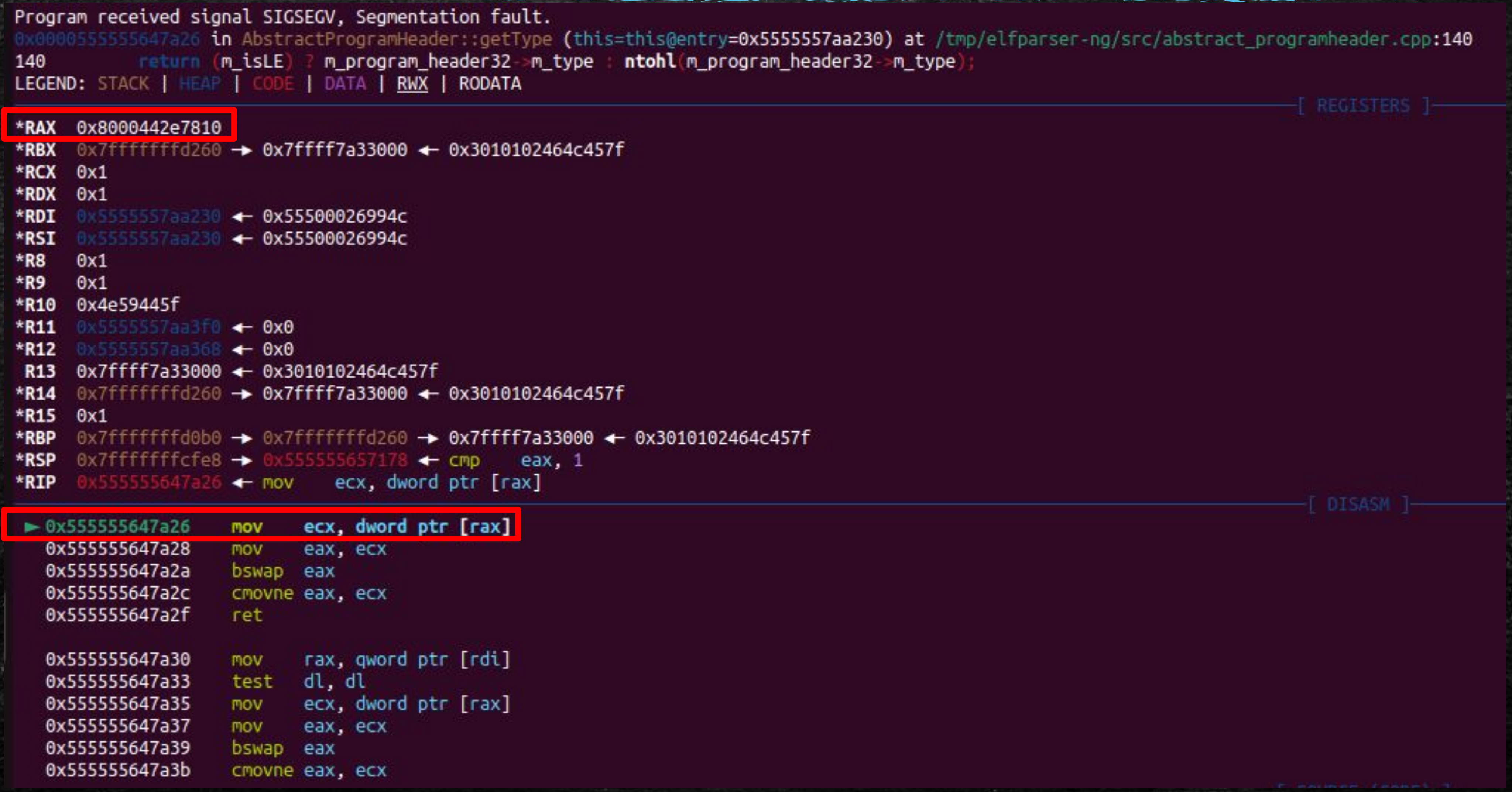
```
►     m_programHeader.setHeaders(ptrDataMem +
                                 m_elfHeader.getProgramOffset(),
                                 m_elfHeader.getProgramCount(),
                                 m_elfHeader.getProgramSize(),
                                                     m_elfHeader.is64(),
                                                     m_elfHeader.isLE());

00:0000│ rsp 0x7fffffffd0e0 ← 0x14061
01:0008│     0x7fffffffd0e8 → 0x7fffffffd128 → 0x55555573d340 ← '/home/xiaobye/Documents/fuzzing_test/el
02:0010│     0x7fffffffd0f0 → 0x7fffffffdd08 → 0x55555573d540 → 0x55555573cbb0 → 0x55555573cbc0 ← ...
03:0018│     0x7fffffffd0f8 → 0x7fffffffd208 ← 0x140615573c0a1
04:0020│ r15 0x7fffffffd100 ← 0x0
05:0028│     0x7fffffffd108 ← 0x0
06:0030│     0x7fffffffd110 ← 0x14061
07:0038│     0x7fffffffd118 ← 0x0

 ► f 0   0x5555555b46c7
   f 1   0x55555558e6a4
   f 2   0x555555587629 main+1769
   f 3   0x7ffff7a75d90 __libc_start_call_main+128
   f 4   0x7ffff7a75e40 __libc_start_main+128
   f 5   0x55555558ba15 _start+37

pwndbg> p ptrDataMem
$1 = 0x7ffff7a33000 "\177ELF\002\001\001\003"
pwndbg> call m_elfHeader.getProgramOffset()
$2 = 1284196368
pwndbg> call m_elfHeader.getProgramOffset()
$3 = 1284196368
pwndbg> pi hex(1284196368)
'0x4c8b4810'
pwndbg> pi hex(1284196368 + 0x7ffff7a33000)
'0x8000442e7810'
pwndbg>
```

Details about bugs



```
Program received signal SIGSEGV, Segmentation fault.
0x0000555555647a26 in AbstractProgramHeader::getType (this=this@entry=0x5555557aa230) at /tmp/elfparser-ng/src/abstract_programheader.cpp:140
140         return (m_isLE) ? m_program_header32->m_type : ntohl(m_program_header32->m_type);
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
─────────────────────────────────────────────────────────────────────[ REGISTERS ]──────
*RAX  0x8000442e7810
*RBX  0x7fffffffd260 → 0x7ffff7a33000 ← 0x3010102464c457f
*RCX  0x1
*RDX  0x1
*RDI  0x5555557aa230 ← 0x55500026994c
*RSI  0x5555557aa230 ← 0x55500026994c
*R8   0x1
*R9   0x1
*R10  0x4e59445f
*R11  0x5555557aa3f0 ← 0x0
*R12  0x5555557aa368 ← 0x0
 R13  0x7ffff7a33000 ← 0x3010102464c457f
*R14  0x7fffffffd260 → 0x7ffff7a33000 ← 0x3010102464c457f
*R15  0x1
*RBP  0x7fffffffd0b0 → 0x7fffffffd260 → 0x7ffff7a33000 ← 0x3010102464c457f
*RSP  0x7fffffffcfe8 → 0x555555657178 ← cmp    eax, 1
*RIP  0x555555647a26 ← mov    ecx, dword ptr [rax]
──────────────────────────────────────────────────────────────────────[ DISASM ]──────
 ► 0x555555647a26    mov    ecx, dword ptr [rax]
   0x555555647a28    mov    eax, ecx
   0x555555647a2a    bswap  eax
   0x555555647a2c    cmovne eax, ecx
   0x555555647a2f    ret

   0x555555647a30    mov    rax, qword ptr [rdi]
   0x555555647a33    test   dl, dl
   0x555555647a35    mov    ecx, dword ptr [rax]
   0x555555647a37    mov    eax, ecx
   0x555555647a39    bswap  eax
   0x555555647a3b    cmovne eax, ecx
```

END