華中科技大學
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

网络空间安全学院

Linux ELF 二进制破解演示

网络空间安全学院 慕冬亮
Email : dzm91@hust.edu.cn

ELF文件格式

```
ecs-assist-user@iZbp1adyuoe4hpyhzlvk3yZ:~/s2_demo/test_cracked_elf$ make
gcc -no-pie -o test_elf test_elf.c
ecs-assist-user@iZbp1adyuoe4hpyhzlvk3yZ:~/s2_demo/test_cracked_elf$ file test_elf
test_elf: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
 interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=d8857b80e206c71dbf54851cdf
19127d2c1c927b, for GNU/Linux 3.2.0, not stripped
ecs-assist-user@iZbp1adyuoe4hpyhzlvk3yZ:~/s2_demo/test_cracked_elf$ readelf -h tes
t_elf
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x401070
  Start of program headers:          64 (bytes into file)
  Start of section headers:          13960 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         13
  Size of section headers:           64 (bytes)
  Number of section headers:         31
  Section header string table index: 30
```

ELF文件格式

```
ecs-assist-user@iZbp1adyuoe4hpyhzlvk3yZ:~/s2_demo/test_cracked_elf$ readelf -l test_elf

Elf file type is EXEC (Executable file)
Entry point 0x401070
There are 13 program headers, starting at offset 64

Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000400040 0x0000000000400040
                 0x00000000000002d8 0x00000000000002d8  R      0x8
  INTERP         0x0000000000000318 0x0000000000400318 0x0000000000400318
                 0x000000000000001c 0x000000000000001c  R      0x1
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
  LOAD           0x0000000000000000 0x0000000000400000 0x0000000000400000
                 0x0000000000000538 0x0000000000000538  R      0x1000
  LOAD           0x0000000000001000 0x0000000000401000 0x0000000000401000
                 0x00000000000001e9 0x00000000000001e9  R E    0x1000
  LOAD           0x0000000000002000 0x0000000000402000 0x0000000000402000
                 0x0000000000000124 0x0000000000000124  R      0x1000
  LOAD           0x0000000000002e10 0x0000000000403e10 0x0000000000403e10
                 0x0000000000000228 0x0000000000000230  RW     0x1000
  DYNAMIC        0x0000000000002e20 0x0000000000403e20 0x0000000000403e20
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  NOTE           0x0000000000000338 0x0000000000400338 0x0000000000400338
                 0x0000000000000030 0x0000000000000030  R      0x8
  NOTE           0x0000000000000368 0x0000000000400368 0x0000000000400368
                 0x0000000000000044 0x0000000000000044  R      0x4
  GNU_PROPERTY   0x0000000000000338 0x0000000000400338 0x0000000000400338
                 0x0000000000000030 0x0000000000000030  R      0x8
  GNU_EH_FRAME   0x000000000000204c 0x000000000040204c 0x000000000040204c
                 0x0000000000000034 0x0000000000000034  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x10
  GNU_RELRO      0x0000000000002e10 0x0000000000403e10 0x0000000000403e10
                 0x00000000000001f0 0x00000000000001f0  R      0x1
```

ELF文件格式

```
ecs-assist-user@iZbp1adyuoe4hpyhzlvk3yZ:~/s2_demo/test_cracked_elf$ readelf -S test_elf
There are 31 section headers, starting at offset 0x3688:

Section Headers:
  [Nr] Name              Type             Address           Offset
       Size              EntSize          Flags  Link  Info  Align
  [ 0]                   NULL             0000000000000000  00000000
       0000000000000000  0000000000000000           0     0     0
  [ 1] .interp           PROGBITS         0000000000400318  00000318
       000000000000001c  0000000000000000   A       0     0     1
  [ 2] .note.gnu.pr[...] NOTE             0000000000400338  00000338
       0000000000000030  0000000000000000   A       0     0     8
  [ 3] .note.gnu.bu[...] NOTE             0000000000400368  00000368
       0000000000000024  0000000000000000   A       0     0     4
  [ 4] .note.ABI-tag     NOTE             000000000040038c  0000038c
       0000000000000020  0000000000000000   A       0     0     4
```

```
  [12] .init             PROGBITS         0000000000401000  00001000
       000000000000001b  0000000000000000  AX       0     0     4
  [13] .plt              PROGBITS         0000000000401020  00001020
       0000000000000030  0000000000000010  AX       0     0     16
  [14] .plt.sec          PROGBITS         0000000000401050  00001050
       0000000000000020  0000000000000010  AX       0     0     16
  [15] .text             PROGBITS         0000000000401070  00001070
       0000000000000169  0000000000000000  AX       0     0     16
  [16] .fini             PROGBITS         00000000004011dc  000011dc
       000000000000000d  0000000000000000  AX       0     0     4
  [17] .rodata           PROGBITS         0000000000402000  00002000
       000000000000004b  0000000000000000   A       0     0     4
```

计算需要修改的二级制代码

objdump -d -M intel test_elf > objdump_test_elf

```
#include <stdio.h>
#include <stdbool.h>
#include <errno.h>

int main()
{
        int result, input;

        printf("3 + 5 = ?\n");
        input = getchar();
        if (input == EOF) return -EINVAL;

        if (input >= '0' && input <= '9') {
                result = input - '0';
        } else {
                printf("Not an integer\n");
        }

        if (result == 8) {
                printf("The answer is correct\n");
        } else {
                printf("The answer is incorrect\n");
        }

        return 0;
}
```

```
83 7d fc 08             cmp    DWORD PTR [rbp-0x4],0x8
75 11                   jne    40119f <main+0x69>
48 8d 05 88 0e 00 00    lea    rax,[rip+0xe88]        #
48 89 c7                mov    rdi,rax
e8 93 fe ff ff          call   401030 <puts@plt>
eb 0f                   jmp    4011ae <main+0x78>
48 8d 05 8d 0e 00 00    lea    rax,[rip+0xe8d]        #
48 89 c7                mov    rdi,rax
e8 82 fe ff ff          call   401030 <puts@plt>
```

计算需要修改的二级制代码

```
401188:       83 7d fc 08             cmp    DWORD PTR [rbp-0x4],0x8
40118c:       75 11                   jne    40119f <main+0x69>
40118e:       48 8d 05 88 0e 00 00    lea    rax,[rip+0xe88]        # 40201d <_IO_stdin_used+0x1d>
```

```
[14] .text             PROGBITS         0000000000401050  00001050
     0000000000000165  0000000000000000  AX       0     0     16
```

X – 0x1050 = 0x40118c – 0x401050

计算可得： X = 0x118c

具体演示

使用hexedit修改二进制文件0x11b0处的二进制代码，怎么修改呢？我们直接把jne翻转变成je即可。而对应到机器码的时候，就是从0x75化成0x74

hexedit test_elf

```
00000000   7F 45 4C 46  02 01 01 00  00 00 00 00  00 00 00 00  02
00000014   01 00 00 00  70 10 40 00  00 00 00 00  40 00 00 00  00
00000028   88 36 00 00  00 00 00 00  00 00 00 00  40 00 38 00  0D
0000003C   1F 00 1E 00  06 00 00 00  04 00 00 00  40 00 00 00  00
00000050   40 00 40 00  00 00 00 00  40 00 40 00  00 00 00 00  D8
00000064   00 00 00 00  D8 02 00 00  00 00 00 00  08 00 00 00  00
00000078   03 00 00 00  04 00 00 00  18 03 00 00  00 00 00 00  18
0000008C   00 00 00 00  18 03 40 00  00 00 00 00  1C 00 00 00  00
000000A0   1C 00 00 00  00 00 00 00  01 00 00 00  00 00 00 00  01
000000B4   04 00 00 00  00 00 00 00  00 00 00 00  00 00 40 00  00

                                                 New position ? 0x11b0
```

快捷键:
Ctrl+g 跳转到对应位置
Ctrl+x 保存并退出

```
0000116C   E8 DF FE FF  FF E8 EA FE  FF FF 89 45
00001180   EA FF FF FF  EB 51 83 7D  FC 2F 7E 11
00001194   FC 83 E8 30  89 45 F8 EB  0F 48 8D 05
000011A8   A4 FE FF FF  83 7D F8 08  75 11 48 8D
```

https://gitee.com/mudongliang/s2_demo

具体演示

```
ecs-assist-user@iZbp1adyuoe4hpyhzlvk3yZ:~/s2_demo/test_cracked_elf$ ./test_elf
3 + 5 = ?
8
The answer is correct
ecs-assist-user@iZbp1adyuoe4hpyhzlvk3yZ:~/s2_demo/test_cracked_elf$ ./test_elf_1
3 + 5 = ?
2
The answer is correct
```