Task2. Patching

Способ 1.

Вывод функции bubble_sort:

```
incomprehensible@incomprehensible:~/0xygenSoftw$ vim task2.c
incomprehensible@incomprehensible:~/0xygenSoftw$ gcc -g -00 task2.c -o task2
incomprehensible@incomprehensible:~/0xygenSoftw$ ./task2
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After sorting: -500 -100 -50 -5 0 0 1 1 90 234 700 12345
incomprehensible@incomprehensible:~/0xygenSoftw$ arm-linux-gnueabi-gcc -g task2.c -o task2
incomprehensible@incomprehensible:~/0xygenSoftw$ ls
exp task1 task1.c task2 task2.c task2_nobubble.c Tasks
```

Импортирую бинарник task2 в Ghidra в формате Raw binary. Нахожу функцию bubble_sort и ее адрес.

```
FUN 000005a0
000005a0 00 48 2d e9
                         stmdh
                                      sp!,{ rll lr }
000005a4 04 b0 8d e2
                          add
                                      rll,sp,#0x4
000005a8 18 d0 4d e2
                                      sp, sp, #0x18
                          sub
000005ac 18 00 0b e5
                          str
                                      r0,[r11,#local_1c]
000005b0 1c 10 0b e5
                                      rl,[rll,#local 20]
                          str
000005b4 00 30 a0 e3
                          mov
                                      r3,#0x0
000005b8 14 30 0b e5
                                      r3,[r11,#local 18]
                          str
000005bc 34 00 00 ea
                                      LAB_00000694
                     LAB 000005c0
                                                                       XREF[1]:
                                                                                     000006
000005c0 00 30 a0 e3
                                      r3.#0x0
                          mov
000005c4 10 30 0b e5
                                      r3,[r11,#local 14]
                          str
000005c8 27 00 00 ea
                                      LAB_0000066c
                     LAB_000005cc
                                                                       XREF[1]:
                                                                                     000006
000005cc 10 30 1b e5
                                     r3,[r11,#local_14]
```

A также функцию multiply_by8 на замену.

```
FUN 000004cc
                                                                      XREF[1]:
                                                                                   FUN
000004cc_00 48 2d e9
                         stmdb
                                     sp!,{ rll lr }
000004d0 04 b0 8d e2
                         add
                                     rll,sp,#0x4
000004d4 10 d0 4d e2
                                     sp,sp,#0x10
                         sub
000004d8 10 00 0b e5
                                     r0,[r11,#local 14]
                         str
000004dc 14 10 0b e5
                         str
                                     rl,[rll,#local_18]
000004e0 00 30 a0 e3
                                     r3,#0x0
                         mov
000004e4 0c 30 0b e5
                                     r3,[r11,#local_10]
                         str
000004e8 0d 00 00 ea
                                     LAB 00000524
                     LAB 000004ec
                                                                      XREF[1]:
                                                                                   0000
000004ec Oc 30 1b e5
                         ldr
                                     r3,[rl1,#local_10]
000004f0 03 31 a0 el
                                     r3, r3, lsl #0x2
                         mov
000004f4 10 20 1b e5
                         ldr
                                     r2,[rl1,#local 14]
000004f8 03 30 82 e0
                         add
                                     r3, r2, r3
000004fc 00 20 93 e5
                         ldr
                                     r2,[r3,#0x0]
00000500 Oc 30 1b e5
                        ldr
                                     r3,[rll,#local_10]
00000504 03 31 a0 el
                         mov
                                     r3,r3, lsl #0x2
00000508 10 10 1b e5
                         ldr
                                     rl,[rll,#local 14]
0000050c 03 30 81 e0
                         add
                                     r3.rl.r3
```

Нахожу функцию main.

```
2 undefined4 FUN 09090719(void)
                r12!,{r0 r1 r2 r3}=>local
                lr,{r0 r1 r2 r3 }=>DAT 000108c4
r12.{r0 r1 r2 r3 }=>local_1c
r3.#0wc
ldmia
stmia
                                                                                                                                                                                            undefined4 uWarl-
                                                                                                                                                                                            int local 44;
undefined4 local 3d (4):
nov
                r3, [r11.#local_49]
r0, [DAT_000037f0]
FUN_00003394
str
                                                                                                                                                                                             undefined4 local 2s:
                                                                                                                                                                                            undefined4 uStack49;
undefined4 uStack39;
                                                                                     undefined FUN 00000394()
ьı
                p3 #3m3
               r3, [r11,elocal_44]
LAB_00000798
                                                                                                                                                                                            undefined4 uStack32:
                                                                                                                                                                                            undefined4 local lo:
undefined4 uStack24:
0000076c
                                                              )FEF[1]:
                                                                                90303744(1)
                                                                                                                                                                                            undefined4 uStock29;
undefined4 uStock15;
                r3.1r11.#local 441
ldr
                r3, r3, lsl #0x2
r2, r11, #0x4
                                                                                                                                                                                            int local c:
                                                                                                                                                                                           add
                 r3, [r3,#-0x34]
CDV
                 r8, IDAT 989897141
                                                                                     = 963138701
                FUN 00000394
r3, [r11.#local 44]
                                                                                     undefined FUN 00000394()
ldr
add
                 r3. r3. e8x1
                r3, [r11,#local_44]
str
00000798
                                                             XREF[1]: 00000768(1)
                r2, [r11, #local_44]
r3, [r11, #local_40]
ldr
ldr
cmp
blt
                                                                                                                                                                                          ustack.6 = DAT GB0807et18ct,

PLV B080804(DAT_GB0807f9),

local_44 = 0,

white? true ) (

if (8co * local_44) ordae:

PLV 8050804(DAT_GB0807f4, local_3c)[local_44]

local_44 = local_44 = 1:
                LAS 0000075c
                r0,#3xa
FUN 000003c4
r3,r11,#3x38
nov
                                                                                     undefined FUN_000003c4()
sub
Lde
                rl. [rll.#local 49]
                r0, r3
FUN 000004cc
bì
                                                                                                                                                                                          ;
FUN_9090954(19);
FUN_909094cc(test_Sr, 9xc);
UNSTL = 9;
If (local_c |= +0NT_909097c8) {
                r2, [DAT_000007c8]
                r1, [r2,#0x0]=*OAT_000108d4
r2, [r11,#local_c]
ldr
                                                                                                                                                                                     41 if (local
                                                                                                                                                                                                FUN_909093±011
cors
                LAB 0000074c
                                                                                                                                                                                            return uWarl:
                                                                                     undefined FUN 000003:0()
                                                             XBEE[1]: 90909744(1)
9898974c
                sp,r11,#9x4
```

Нахожу инструкцию, где вызывается функция bubble_sort и дизассемблирую.

```
υL
                    FUN 000003C4
eu
                    r3, r11, #0x38
e2
       sub
e5
       ldr
                    rl,[rll,#local 40]
el
                    r0, r3
       cpy
eb
       bl
                    0x000005a0
е3
       mov
                    r3,#0x0
       ldr
e5
                    r2, [DAT 000007e8]
e5
       ldr
                    r1, [r2, \#0x0] => DAT 000108d4
                    r2.[rll.#local c]
e5
       ldr
```

Вставляю в операнд инструкции перехода bl адрес первой инструкции функции multiply_by8. Готово.

Полученный бинарник сохраняю как task2.bin.

Копирую пропатченную программу на Raspberry pi и тестирую (new_task2).

```
pi@raspberrypi:~ $ ls
BigBro abbro bench new_task2
pi@raspberrypi:~ $ chmod +x new_task2
pi@raspberrypi:~ $ ./new_task2
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After multiplying: 8 720 -4000 0 0 8 1872 98760 5600 -400 -40 -800
pi@raspberrypi:~ $ []
```

Отработала multiply_by8.

Способ 2.

Я выделила свои функции в обоих исходниках в отдельную секцию. Назвала ее .patch.

```
#include <stdio.h>
void bubble_sort() __attribute__((section (".patch")));
void bubble_sort(int *arr, int sz)
        int tmp;
        for (int i = 0; i < sz; ++i)</pre>
                for (int j = 0; j < sz - i - 1; + + j)
                         if (arr[j] > arr[j + 1])
                                 tmp = arr[j];
                                 arr[j] = arr[j+1];
arr[j+1] = tmp;
int main()
        int arr[] = { 1, 90, -500, 0, 0, 1, 234, 12345, 700, -50, -5, -100};
int sz = sizeof(arr) / sizeof(int);
        putchar('\n');
        bubble_sort(&arr[0], sz);
        putchar('\n');
return 0;
```

У меня есть два executable файла.

Их вывод на Raspberry pi:

```
pi@raspberrypi:~ $ ls
BigBro abbro bench task2 task2_no
pi@raspberrypi:~ $ ./task2 #bubble_sort
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After routine: -500 -100 -50 -5 0 0 1 1 90 234 700 12345
pi@raspberrypi:~ $ ./task2_no #multiply by 8
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After routine: 8 720 -4000 0 0 8 1872 98760 5600 -400 -40 -800
pi@raspberrypi:~ $ []
```

Теперь я могу вытащить секцию .patch из нужного мне бинарника (task2_no), который содержит multiply_by8. Сохраняю ее как patch.bin.

И заменяю секцию task2, содержащего пузырьковую сортировку, на patch.bin. При тестировании я не указывала новый файл в output и заменила сам executable task2. Но в готовой версии создала новый файл task2_new.

```
incomprehensible@incomprehensible: ~/Courses/IP... × incomprehensible@incomprehensible: ~/OxygenSoft... × incomprehensible@incomprehensible: ~/OxygenSoftw/exp$ ls task2 task2.c task2_no task2_nobubble.c task2_no_sections task2_sections incomprehensible@incomprehensible: ~/OxygenSoftw/exp$ arm-linux-gnueabi-objcopy -j.patch -Obinary task2_no patch.bin incomprehensible@incomprehensible: ~/OxygenSoftw/exp$ arm-linux-gnueabi-objcopy --update-section .patch=patch.bin task2 incomprehensible@incomprehensible: ~/OxygenSoftw/exp$ scp task2 pi@192.168.1.40:/home/pi/task2_new pi@192.168.1.40's password: task2 incomprehensible@incomprehensible: ~/OxygenSoftw/exp$
```

Переношу исполняемый файл на свою Raspberry рі для быстрой проверки (task2_new).

```
pi@raspberrypi:~ $ ls
BigBro abbro bench task2 task2_no
pi@raspberrypi:~ $ ./task2 #bubble_sort
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After routine: -500 -100 -50 -5 0 0 1 1 90 234 700 12345
pi@raspberrypi:~ $ ./task2_no #multiply by 8
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After routine: 8 720 -4000 0 0 8 1872 98760 5600 -400 -40 -800
pi@raspberrypi:~ $ ls
BigBro abbro bench task2 task2_new task2_no
pi@raspberrypi:~ $ ./task2_new #patched
Before routine: 1 90 -500 0 0 1 234 12345 700 -50 -5 -100
After routine: 8 720 -4000 0 0 8 1872 98760 5600 -400 -40 -800
pi@raspberrypi:~ $ ]
```

Видим, что замена произошла.