南開大學
Nankai University

计算机学院

汇编语言与逆向技术实验报告

实验 8 CTF（Capture The Flag）夺旗赛

姓名：杨冰雪

学号：2110508

专业：计算机科学与技术

2023 年 12 月 22 日

目录

1 实验目的

1. 熟悉静态反汇编工具 IDA Pro；
2. 熟悉动态反汇编工具 OllyDbg；
3. 掌握对二进制代码内部逻辑关系的分析；
4. 掌握对二进制代码的修改和保存。

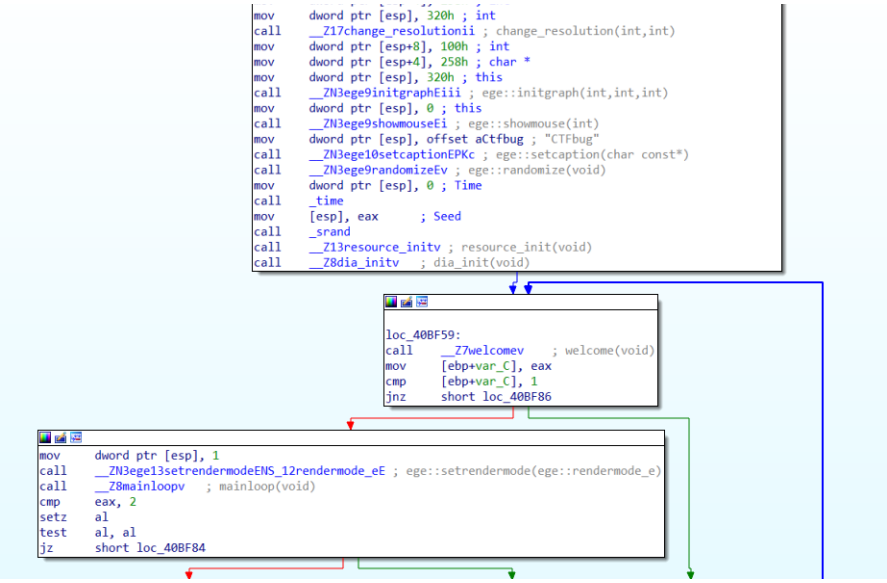
2 实验内容

2.1 逆向分析

逆向分析 game.exe 二进制代码的主要逻辑结构和重要数据。

2.1.1 主要逻辑结构

main 函数 首先在 main 函数中，调用了一系列函数，用于对声音、图像等的初始化。此外还调用了 welcome 函数，进入游戏的欢迎界面，然后调用循环函数 mainloop，进行游戏画面的切换和关闭。



mainloop 函数 在 mainloop 函数中，调用了许多重要的函数，例如：

- 展示当前游戏的关卡进度

```
ext:004070CE loc_4070CE:                        ; CODE XREF: mainloop(void)+623↑j
ext:004070CE                mov     dword ptr [esp+4], 1 ; bool
ext:004070D6                mov     dword ptr [esp], offset aZKeyDecrypting ; "Z KEY DECRYPTING PROGRESS : 0%%"
ext:004070DD                mov     [ebp+fctx.call_site], 0FFFFFFFFh
ext:004070E7                call    __Z9show_textPcb ; show_text(char *,bool)
ext:004070EC                jmp     loc_407174
ext:004070F1 ; ---------------------------------------------------------------------------
ext:004070F1
ext:004070F1 loc_4070F1:                        ; CODE XREF: mainloop(void)+619↑j
ext:004070F1                mov     dword ptr [esp+4], 1 ; int
ext:004070F9                mov     dword ptr [esp], offset aZKeyDecrypting_0 ; "Z KEY DECRYPTING PROGRESS : 25%%"
ext:00407100                mov     [ebp+fctx.call_site], 0FFFFFFFFh
ext:0040710A                call    __Z9show_textPcb ; show_text(char *,bool)
ext:0040710F                mov     dword ptr [esp], 1 ; this
ext:00407116                call    __ZN3KEY8writekeyEi ; KEY::writekey(int)
ext:0040711B                jmp     short loc_407174
ext:0040711D ; ---------------------------------------------------------------------------
ext:0040711D
ext:0040711D loc_40711D:                        ; CODE XREF: mainloop(void)+62D↑j
ext:0040711D                mov     dword ptr [esp+4], 1 ; int
ext:00407125                mov     dword ptr [esp], offset aZKeyDecrypting_1 ; "Z KEY DECRYPTING PROGRESS : 50%%"
ext:0040712C                mov     [ebp+fctx.call_site], 0FFFFFFFFh
ext:00407136                call    __Z9show_textPcb ; show_text(char *,bool)
ext:0040713B                mov     dword ptr [esp], 2 ; this
ext:00407142                call    __ZN3KEY8writekeyEi ; KEY::writekey(int)
ext:00407147                jmp     short loc_407174
ext:00407149 ; ---------------------------------------------------------------------------
ext:00407149
ext:00407149 loc_407149:                        ; CODE XREF: mainloop(void)+632↑j
ext:00407149                mov     dword ptr [esp+4], 1 ; int
ext:00407151                mov     dword ptr [esp], offset aZKeyDecrypting_2 ; "Z KEY DECRYPTING PROGRESS : 75%%"
ext:00407158                mov     [ebp+fctx.call_site], 0FFFFFFFFh
ext:00407162                call    __Z9show_textPcb ; show_text(char *,bool)
ext:00407167                mov     dword ptr [esp], 3 ; this
ext:0040716E                call    __ZN3KEY8writekeyEi ; KEY::writekey(int)
```

- 当捡拾到道具时，生命值会进行恢复

```
ext:004071AD loc_4071AD:                        ; CODE XREF: mainloop(void)+2ED↑j
ext:004071AD                mov     eax, [ebp+var_20]
ext:004071B0                imul    eax, 0A8h
ext:004071B6                add     eax, offset unk_52E7A0
ext:004071BB                mov     eax, [eax+8]
ext:004071BE                mov     [esp+8], eax
ext:004071C2                mov     dword ptr [esp+4], offset aYourHealthReco ; "Your health recovered %d."
ext:004071CA                lea     eax, [ebp+Buffer]
ext:004071D0                mov     [esp], eax      ; Buffer
ext:004071D3                call    _sprintf
```

- 当捡拾到道具时，会获得钻石

```
ext:00407278 loc_407278:                        ; CODE XREF: mainloop(void)+2F6↑j
ext:00407278                mov     eax, [ebp+var_20]
ext:0040727B                imul    eax, 0A8h
ext:00407281                add     eax, offset unk_52E7A0
ext:00407286                mov     eax, [eax+8]
ext:00407289                mov     [esp+8], eax
ext:0040728D                mov     dword ptr [esp+4], offset aYouGotDDiamond ; "You got %d diamonds."
ext:00407295                lea     eax, [ebp+Buffer]
ext:0040729B                mov     [esp], eax      ; Buffer
ext:0040729E                call    _sprintf
```

- 切换关卡时，会保存人物的血量、技能等属性

```
ext:00407174 loc_407174:                        ; CODE XREF: mainloop(void)+625↑j
ext:00407174                                    ; mainloop(void)+638↑j ...
ext:00407174                mov     dword ptr [esp], offset _before_level
ext:0040717B                call    __Z4saveR8savedata ; save(savedata &)
ext:00407180                mov     eax, [ebp+var_20]
ext:00407183                imul    eax, 0A8h
ext:00407189                add     eax, offset unk_52E790
ext:0040718E                mov     eax, [eax+8]
ext:00407191                mov     [esp], eax      ; int
ext:00407194                mov     [ebp+fctx.call_site], 0FFFFFFFFh
ext:0040719E                call    __Z11prepare_mapi ; prepare_map(int)
ext:004071A3                call    __Z10logic_initv ; logic_init(void)
ext:004071A8                jmp     loc_407314
```

2.1.2 重要数据

在数据段保存了许多重要的数据，例如人物血量、人物移动速度、击杀怪物数量、怪物血量等。

```
.data:004E0048 _MOVE_SPEED     dd 3.1415925            ; DATA XREF: mainloop(void)+12B7↑r
.data:004E0048                                         ; mainloop(void)+12D6↑r ...
.data:004E004C                 public _MAX_HP
.data:004E004C _MAX_HP         dd 43960000h            ; DATA XREF: save(savedata &)+18↑r
.data:004E004C                                         ; apply_save(savedata)+1E↑w ...
.data:004E0050                 public _ARMOR
.data:004E0050 _ARMOR          dd 41200000h            ; DATA XREF: save(savedata &)+4A↑r
.data:004E0050                                         ; apply_save(savedata)+44↑w ...
.data:004E0054                 public _spawnX
.data:004E0054 ; float spawnX
.data:004E0054 _spawnX         dd 41200000h            ; DATA XREF: logic_init(void)+A5↑r
.data:004E0054                                         ; mainloop(void)+5DC↑w
.data:004E0058                 public _spawnY
.data:004E0058 ; float spawnY
.data:004E0058 _spawnY         dd 43960000h            ; DATA XREF: logic_init(void)+9F↑r
.data:004E0058                                         ; mainloop(void)+5FF↑w
.data:004E005C                 public _csize_x
.data:004E005C ; float csize_x
.data:004E005C _csize_x        dd 32.0                 ; DATA XREF: mainloop(void)+1E2↑r
.data:004E005C                                         ; mainloop(void)+271↑r ...
.data:004E0060                 public _csize_y
.data:004E0060 ; float csize_y
.data:004E0060 _csize_y        dd 42400000h            ; DATA XREF: mainloop(void)+1DC↑r
.data:004E0060                                         ; mainloop(void)+26B↑r ...
.data:004E0064                 public _last_combat
.data:004E0064 _last_combat    dd 0FFFFFF9Ch           ; DATA XREF: logic_init(void)+35↑w
.data:004E0064                                         ; mainloop(void)+16FC↑r ...
.data:004E0068                 public _FIRE_SPEED
.data:004E0068 _FIRE_SPEED     dd 8.0                  ; DATA XREF: shot(float,float,float,float,int)+1D6↑r
.data:004E0068                                         ; shot(float,float,float,float,int)+1FE↑r
.data:004E006C                 public _FIRE_SCOPE
.data:004E006C _FIRE_SCOPE     dd 800000.0             ; DATA XREF: mainloop(void)+2A4B↑r
.data:004E0070                 public _INITIAL_HP
.data:004E0070 _INITIAL_HP     dd 43960000h            ; DATA XREF: data_init(void)+6↑r
.data:004E0074                 public _SLOW_DEG
.data:004E0074 _SLOW_DEG       dd 1.0                  ; DATA XREF: mainloop(void)+2570↑r
.data:004E0074                                         ; mainloop(void)+258C↑r ...
.data:004E0078                 public _cur_map
.data:004E0078 _cur_map        dd 1                    ; DATA XREF: save(savedata &)+5D↑r
.data:004E0078                                         ; prepare_map(int)+9C↑w
```

2.2 代码修改

2.2.1 基本思路

1. 在开始游戏后，人物很容易碰到怪兽而导致血量不足死亡

   解决方法：把人物血量设为无穷大，就不会死亡了

2. 游戏通过条件是‘You need to kill enough monsters!’

   解决方法：修改通过条件，变为无条件通关

2.2.2 修改血量

找到代码段修改人物血量，将人物的最大血量和初始血量都设为最大

```
.data:004E004C                 public _MAX_HP
.data:004E004C _MAX_HP         dd 0FFFFFFFFh           ; DATA XREF: save(savedata &)+18↑r
.data:004E004C                                         ; apply_save(savedata)+1E↑w ...
.data:004E0050                 public _ARMOR
.data:004E0050 _ARMOR          dd 41200000h            ; DATA XREF: save(savedata &)+4A↑r
.data:004E0050                                         ; apply_save(savedata)+44↑w ...
.data:004E0054                 public _spawnX
.data:004E0054 ; float spawnX
.data:004E0054 _spawnX         dd 41200000h            ; DATA XREF: logic_init(void)+A5↑r
.data:004E0054                                         ; mainloop(void)+5DC↑w
.data:004E0058                 public _spawnY
.data:004E0058 ; float spawnY
.data:004E0058 _spawnY         dd 43960000h            ; DATA XREF: logic_init(void)+9F↑r
.data:004E0058                                         ; mainloop(void)+5FF↑w
.data:004E005C                 public _csize_x
.data:004E005C ; float csize_x
.data:004E005C _csize_x        dd 32.0                 ; DATA XREF: mainloop(void)+1E2↑r
.data:004E005C                                         ; mainloop(void)+271↑r ...
.data:004E0060                 public _csize_y
.data:004E0060 ; float csize_y
.data:004E0060 _csize_y        dd 42400000h            ; DATA XREF: mainloop(void)+1DC↑r
.data:004E0060                                         ; mainloop(void)+26B↑r ...
.data:004E0064                 public _last_combat
.data:004E0064 _last_combat    dd 0FFFFFF9Ch           ; DATA XREF: logic_init(void)+35↑w
.data:004E0064                                         ; mainloop(void)+16FC↑r ...
.data:004E0068                 public _FIRE_SPEED
.data:004E0068 _FIRE_SPEED     dd 8.0                  ; DATA XREF: shot(float,float,float,float,int)+1D6↑r
.data:004E0068                                         ; shot(float,float,float,float,int)+1FE↑r
.data:004E006C                 public _FIRE_SCOPE
.data:004E006C _FIRE_SCOPE     dd 800000.0             ; DATA XREF: mainloop(void)+2A4B↑r
.data:004E0070                 public _INITIAL_HP
.data:004E0070 _INITIAL_HP     dd 0FFFFFFFFh           ; DATA XREF: data_init(void)+6↑r
.data:004E0074                 public _SLOW_DEG
.data:004E0074 _SLOW_DEG       dd 1.0                  ; DATA XREF: mainloop(void)+2570↑r
.data:004E0074                                         ; mainloop(void)+258C↑r ...
```

2.2.3 修改通关条件

1. 首先通过查询功能找到字符串‘You need to kill enough monsters!’

Address	Function	Instruction	
.text:00406FE7	__Z8mainloopv		mov dword ptr [esp], offset Str ; "You need to kill enough monsters!"
.rdata:004EB440		Str	db 'You need to kill enough monsters!',0

2. 找到其所在的代码段，修改代码为直接跳到下一个

```
.text:00406FDB                 test    al, al
.text:00406FDD                 jz      short loc_406FF8
.text:00406FDF                 mov     dword ptr [esp+4], 3Ch ; '<' ; int
.text:00406FE7                 mov     dword ptr [esp], offset Str ; "You need to kill enough monsters!"
.text:00406FEE                 call    __Z5toastPci    ; toast(char *,int)
.text:00406FF3                 jmp     short loc_406FF8
.text:00406FF3 ; ---------------------------------------------------------------------------
.text:00406FF5                 db    3
.text:00406FF6                 db    0
.text:00406FF7                 db    0
.text:00406FF8 ; ---------------------------------------------------------------------------
.text:00406FF8
.text:00406FF8 loc_406FF8:                             ; CODE XREF: mainloop(void)+54C↑j
.text:00406FF8                                         ; mainloop(void)+562↑j
.text:00406FF8                 mov     dword ptr [esp+8], 320h ; Size
.text:00407000                 mov     dword ptr [esp+4], 0 ; Val
```

3 实验结果

运行被修改的可执行文件，最终得到 Flag

4 实验总结

通过本次实验，让我能够更加熟练的掌握 ida 反汇编代码的分析，学会了对二进制代码的修改和保存，也深刻体会到了逆向分析的有趣之处。