汇编语言与逆向技术实验报告

Lab7- Capture The Flag

学号：2112514　姓名：辛浩然　专业：信息安全、法学

一、 实验步骤

1. 逆向分析 game.exe 二进制代码的主要逻辑结构和重要数据；
2. 修改 game.exe 二进制代码，获得最后的 Flag.

二、 逆向分析

程序执行时，首先执行启动函数。当所有的初始化操作完成后，启动函数调用 main 函数。首先进行了一些背景音乐、图形、渲染等初始化设置，随后调用 mainloop 函数。

在 mainloop 函数中，调用 data_init 函数进行初始化数据。在该函数中，

- 首先获取初始血量，赋值给当前血量。查看关于血量的数据，_INITIAL_HP 为初始血量，HP 为血量。_MAX_HP 为最大血量。
- 将弹药赋值为 14h，_bullets 为弹药数量，武器的使用具有血量限制。查看关于武器的相关数据，_weapons_total 为武器的数量，其值为 4._cur_weapon 为现在的武器序号。
- 调用 savedata 函数保存数据。

```
mov     ds:dword_4FDD44, 1
mov     ds:_bullets, 14h
mov     ds:dword_4FDD48, 0
mov     ds:dword_4FDD4C, 5
mov     ds:dword_4FDD50, 0Ah
```

- 对不同武器的需要的弹药数进行初始化。
- 调用 prepare_map 函数，进行地图初始化。
- 随后，调用 logic_init 函数，进行游戏逻辑的初始化。首先，设置一些鼠标操作；然后调用 set_posi 函数，调用 calc_camera 函数进行定位人物等。随后，调用 generate_monster 函数，怪兽的初始化。

```
loc_406FD4:                             ; CODE XREF: mainloop(void)+48E↑j
                                        ; mainloop(void)+4A1↑j ...
        movzx   eax, [ebp+var_1A]
        xor     eax, 1
        test    al, al
        jz      short loc_406FF8
        mov     dword ptr [esp+4], 3Ch ; '<' ; int
        mov     dword ptr [esp], offset Str ; "You need to kill enough monsters!"
        call    __Z5toastPci    ; toast(char *,int)
        jmp     loc_407314
```

```
loc_4070CE:                             ; CODE XREF: mainloop(void)+623↑j
                mov     dword ptr [esp+4], 1 ; bool
                mov     dword ptr [esp], offset aZKeyDecrypting ; "Z KEY DECRYPTING PROGRESS : 0%%"
                mov     [ebp+fctx.call_site], 0FFFFFFFFh
                call    __Z9show_textPcb ; show_text(char *,bool)
                jmp     loc_407174
; ---------------------------------------------------------------------------

loc_4070F1:                             ; CODE XREF: mainloop(void)+619↑j
                mov     dword ptr [esp+4], 1 ; int
                mov     dword ptr [esp], offset aZKeyDecrypting_0 ; "Z KEY DECRYPTING PROGRESS : 25%%"
                mov     [ebp+fctx.call_site], 0FFFFFFFFh
                call    __Z9show_textPcb ; show_text(char *,bool)
                mov     dword ptr [esp], 1 ; this
                call    __ZN3KEY8writekeyEi ; KEY::writekey(int)
                jmp     short loc_407174
; ---------------------------------------------------------------------------

loc_40711D:                             ; CODE XREF: mainloop(void)+62D↑j
                mov     dword ptr [esp+4], 1 ; int
                mov     dword ptr [esp], offset aZKeyDecrypting_1 ; "Z KEY DECRYPTING PROGRESS : 50%%"
                mov     [ebp+fctx.call_site], 0FFFFFFFFh
                call    __Z9show_textPcb ; show_text(char *,bool)
                mov     dword ptr [esp], 2 ; this
                call    __ZN3KEY8writekeyEi ; KEY::writekey(int)
                jmp     short loc_407174
; ---------------------------------------------------------------------------

loc_407149:                             ; CODE XREF: mainloop(void)+632↑j
                mov     dword ptr [esp+4], 1 ; int
                mov     dword ptr [esp], offset aZKeyDecrypting_2 ; "Z KEY DECRYPTING PROGRESS : 75%%"
                mov     [ebp+fctx.call_site], 0FFFFFFFFh
                call    __Z9show_textPcb ; show_text(char *,bool)
                mov     dword ptr [esp], 3 ; this
                call    __ZN3KEY8writekeyEi ; KEY::writekey(int)
                nop
```

- 当角色攻击怪兽，怪兽血量降低，怪兽血量为 0 时被消灭。角色会随机恢复体力和得到钻石。受到怪兽攻击，角色会掉血。消灭完全部怪兽后，显示地图进度，进入下一关，渲染下一张地图、进行初始化。而如果未消灭全部怪兽，则弹出提示，不得进入下一关。
- 共需经过四关。当经过四关后回到初始地图，随后通过左下入海口进入最终地图，聊天后得到 flag.

三、 获取 flag

通过游戏试玩，可以发现，遇到的问题有：角色血量太低、弹药限制。

1.首先**修改角色的血量**。crtl+s 进入.data 段，找到_MAX_HP 和_INITIAL_HP，分别代表最大血量和初始血量。

```
.data:004E0070                  public _INITIAL_HP
.data:004E0070 _INITIAL_HP      dd 43960000h

.data:004E004C                  public _MAX_HP
.data:004E004C _MAX_HP          dd 43960000h
```

右击数据名，选择 Jump in a new hex window，打开十六进制视图，找到要修改的数据。

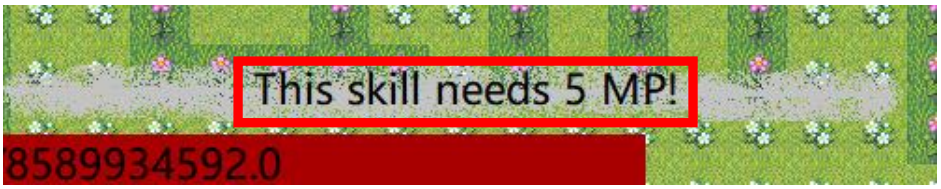
```
A1 39 CD 0D 28 EA  >....
69 6F C8 53 44 39  .....
49 40 00 00 96 43  CU]`.
96 43 00 00 00 42  ..·A.
00 41 00 50 43 49  ..@B.
00 00 01 00 00 00     C
```

右击选择 edit，进行数据修改。Apply changes 保存修改。这里将最大血量和初始血量都修改为 50000000h。

```
.data:004E004C                  public _MAX_HP
.data:004E004C _MAX_HP          dd 50000000h

.data:004E0070                  public _INITIAL_HP
.data:004E0070 INITIAL HP       dd 50000000h
```

通过 Edit-Patch program-Apply patches to input file，将修改同步到文件中。打开程序，可以发现，血量修改成功，受到攻击也不会掉血。

2.随后**修改弹药**。通过游戏试玩，发现弹药不足时会弹出 This skill needs 5 MP!对此进修改。



找到相关代码。

```
.text:00407E97                cmp     edx, eax
.text:00407E99                jle     short loc_407EA2
.text:00407E9B                mov     eax, 1
.text:00407EA0                jmp     short loc_407EA7
.text:00407EA2 ; ---------------------------------------------------------------------------
.text:00407EA2
.text:00407EA2 loc_407EA2:                            ; CODE XREF: mainloop(void)+13E2↑j
.text:00407EA2                                        ; mainloop(void)+13F3↑j ...
.text:00407EA2                mov     eax, 0
.text:00407EA7
.text:00407EA7 loc_407EA7:                            ; CODE XREF: mainloop(void)+140F↑j
.text:00407EA7                test    al, al
.text:00407EA9                jz      short loc_407F12
.text:00407EAB                mov     eax, _cur_weapon
.text:00407EB0                mov     eax, ds:_bltcost[eax*4]
.text:00407EB7                mov     [esp+8], eax
.text:00407EBB                mov     dword ptr [esp+4], offset aThisSkillNeeds ; "This skill needs %d MP!"
```

```
loc_407F12:                             ; CODE XREF: mainloop(void)+1418↑j
                lea     eax, [ebp+var_F4]
                mov     ecx, eax
                call    __ZN3ege9mouse_msg7is_downEv ; ege::mouse_msg::is_down(void)
                test    al, al
                jz      short loc_407F50
                lea     eax, [ebp+var_F4]
                mov     ecx, eax
                call    __ZN3ege9mouse_msg7is_leftEv ; ege::mouse_msg::is_left(void)
                test    al, al
                jz      short loc_407F50
                mov     eax, _cur_weapon
                mov     edx, ds:_bltcost[eax*4]
                mov     eax, ds:_bullets
                cmp     edx, eax
                jg      short loc_407F50
                mov     eax, 1
                jmp     short loc_407F55
```

在 407EA7 地址处，首先对 al 自身进行按位与操作。如果为 0，跳转 407F12 地址处，对该处过程进行分析，发现是使用武器的操作；如果不为 0，会弹出弹药不足的提示。这里将跳转 407F12 地址处的指令修改为无条件跳转。通过 Edit-Patch program-Assemble 修改代码，并 Apply patches to input file.

```
loc_407EA7:                             ;
                test    al, al
                jmp     short loc_407F12
```

继续分析该过程，发现在一次武器释放技能后，会进行比较所需弹药与所剩弹药。如果所需大于所剩，跳转 407F50 处，将 eax 置为 0；如果不大于，将 eax 置为 1，跳转 407F55 处。接下来会对 al 自身进行按位与操作，如果为 0，跳转 408916 处。为 0 意味着所需大于所剩，这样无法释放弹药。因此需要修改。

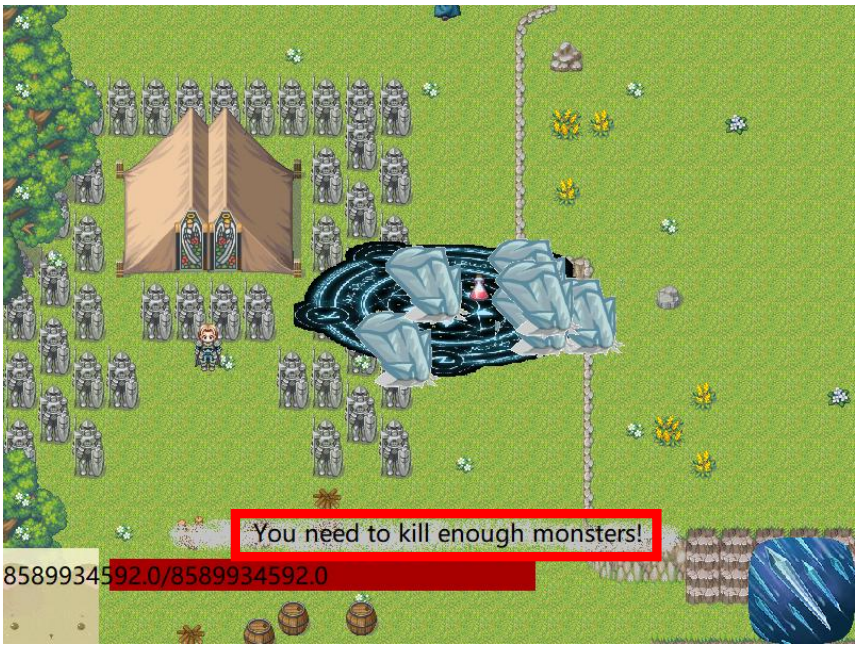
```
                mov     edx, ds:_bltcost[eax*4]
                mov     eax, ds:_bullets
                cmp     edx, eax
                jg      short loc_407F50
                mov     eax, 1
                jmp     short loc_407F55
; ---------------------------------------------------------------------------

loc_407F50:                             ; CODE XREF: mainloop(void)+1490↑j
                                        ; mainloop(void)+14A1↑j ...
                mov     eax, 0

loc_407F55:                             ; CODE XREF: mainloop(void)+14BD↑j
                test    al, al
                jz      loc_408916
                mov     edx, ds:_bullets
                mov     eax, _cur_weapon
                mov     eax, ds:_bltcost[eax*4]
                sub     edx, eax
                mov     eax, edx
                mov     ds:_bullets, eax
```

将 407F50 处指令修改为将 eax 同样置为 1，代码修改为 mov eax,1. 通过 Edit-Patch program-Assemble 修改代码，并 Apply patches to input file.

```
xt:00407F49                 mov     eax, 1
xt:00407F4E                 jmp     short loc_407F55
xt:00407F50 ; ---------------------------------------------------------------------------
xt:00407F50
xt:00407F50 loc_407F50:                             ; CODE XREF: mainloop(void)+1490↑j
xt:00407F50                                         ; mainloop(void)+14A1↑j ...
xt:00407F50                 mov     eax, 1
xt:00407F55
xt:00407F55 loc_407F55:                             ; CODE XREF: mainloop(void)+14BD↑j
xt:00407F55                 test    al, al
xt:00407F57                 jz      loc_408916
```

进入游戏，查看效果。角色确实可以无限制释放武器技能。

3.在修改过后，发现依次杀死怪兽太麻烦，而且有时会找不到怪兽。在怪兽没杀完时，会提示 You need to kill enough monsters!对此进行修改。



找到相关代码。

```
loc_406FD4:                             ; CODE XREF: mainloop(void)+48E↑j
                                        ; mainloop(void)+4A1↑j ...
                movzx   eax, [ebp+var_1A]
                xor     eax, 1
                test    al, al
                jz      short loc_406FF8
                mov     dword ptr [esp+4], 3Ch ; '<' ; int
                mov     dword ptr [esp], offset Str ; "You need to kill enough monsters!"
                call    __Z5toastPci    ; toast(char *,int)
                jmp     loc_407314
```

进行分析，可以发现，对 al 自身按位与，为 0 跳转 406FF8 处；不为 0 则弹出怪兽未杀完的提醒。所以将跳转 406FF8 处更改为无条件跳转。通过 Edit-Patch program-Assemble 修改代码，并 Apply patches to input file.

```
loc_406FD4:                             ; CODE XREF: mainloop(void)+48E↑j
                                        ; mainloop(void)+4A1↑j ...
                movzx   eax, [ebp+var_1A]
                xor     eax, 1
                test    al, al
                jmp     short loc_406FF8
; ---------------------------------------------------------------------------
                mov     dword ptr [esp+4], 3Ch ; '<' ; int
                mov     dword ptr [esp], offset Str ; "You need to kill enough monsters!"
                call    __Z5toastPci    ; toast(char *,int)
                jmp     loc_407314
```

进入游戏，可以不残杀怪兽而通关。

4.经过上述修改，可以顺利通关。得到 flag{a2fdkd80xo}.

四、 心得体会

1.通过对游戏的逆向分析，熟悉了 IDA 的操作，尤其是二进制代码的修改和保存；

2.在分析过程中，提高逆向分析的能力。