

获取僵尸数组

首先 通过模糊搜索得到普通僵尸的血量 (270) 和 普通豌豆射手的攻击力 (20)

从而可以找到一个僵尸的血量地址

	无描述	225EFC48	4 Bytes	959
---	-----	----------	---------	-----

将其修改大 以免一会就死了

```
> 00566D10 - mov [ebp+000000C8],esi
```

ebp+c8 一般是从传递的参数中获取数据 查看详细汇编

1	PlantsVsZombies.exe+166C80	- 83 EC 08	- sub esp,08	函数头
2	PlantsVsZombies.exe+166C83	- 8B 44 24 14	- mov eax,[esp+14]	
3	PlantsVsZombies.exe+166C87	- 53	- push ebx	
4	PlantsVsZombies.exe+166C88	- 55	- push ebp	
5	PlantsVsZombies.exe+166C89	- 8B 6C 24 14	- mov ebp,[esp+14]	
6	PlantsVsZombies.exe+166C8D	- 56	- push esi	
7	PlantsVsZombies.exe+166C8E	- 57	- push edi	
8	PlantsVsZombies.exe+166C8F	- 8B F8	- mov edi,eax	
9	PlantsVsZombies.exe+166C91	- 83 E7 08	- and edi,08	
10	PlantsVsZombies.exe+166C94	- 89 7C 24 10	- mov [esp+10],edi	
11	PlantsVsZombies.exe+166C98	- 75 07	- jne PlantsVsZombies.exe+166C9A	
12	PlantsVsZombies.exe+166C9A	- C7 45 54 19000000	- mov [ebp+54],00000019	
13	PlantsVsZombies.exe+166CA1	- A8 04	- test al,04	
14	PlantsVsZombies.exe+166CA3	- 74 50	- je PlantsVsZombies.exe+166CA5	
15	PlantsVsZombies.exe+166CA5	- 8B F5	- mov esi,ebp	
16	PlantsVsZombies.exe+166CA7	- E8 84060000	- call PlantsVsZombies.exe+166CAB	
17	PlantsVsZombies.exe+166CAC	- 84 C0	- test al,al	
18	PlantsVsZombies.exe+166CAE	- 74 45	- je PlantsVsZombies.exe+166CB0	
19	PlantsVsZombies.exe+166CB0	- 83 BD AC000000 00	- cmp dword ptr [ebp+0000AC],0	
20	PlantsVsZombies.exe+166CB7	- 75 1D	- jne PlantsVsZombies.exe+166CB9	
21	PlantsVsZombies.exe+166CB9	- 8B 45 00	- mov eax,[ebp+00]	
22	PlantsVsZombies.exe+166CBC	- 80 B8 E9090000 00	- cmp byte ptr [eax+00000909],0	
23	PlantsVsZombies.exe+166CC3	- 75 11	- jne PlantsVsZombies.exe+166CC5	
24	PlantsVsZombies.exe+166CC5	- 8B 80 88080000	- mov eax,[eax+00000888]	
25	PlantsVsZombies.exe+166CCB	- 50	- push eax	
26	PlantsVsZombies.exe+166CCC	- BE 12000000	- mov esi,00000012	
27	PlantsVsZombies.exe+166CD1	- E8 4AB9FDFF	- call PlantsVsZombies.exe+166CDE	
28	PlantsVsZombies.exe+166CD6	- 8B 85 AC000000	- mov eax,[ebp+000000AC]	
29	PlantsVsZombies.exe+166CDC	- 3D E8030000	- cmp eax,000003E8	
30	PlantsVsZombies.exe+166CE1	- 7D 05	- jnl PlantsVsZombies.exe+166CE3	

31	PlantsVsZombies.exe+166CE3	- B8 E8030000	- mov eax,000003E8
32	PlantsVsZombies.exe+166CE8	- 8B F5	- mov esi,ebp
33	PlantsVsZombies.exe+166CEA	- 89 85 AC000000	- mov [ebp+000000AC],eax
34	PlantsVsZombies.exe+166CF0	- E8 BBDAFFFF	- call PlantsVsZombies.exe
35	PlantsVsZombies.exe+166CF5	- 8B B5 C8000000	- mov esi,[ebp+000000C8]
36	PlantsVsZombies.exe+166CFB	- 8B C5	- mov eax,ebp
37	PlantsVsZombies.exe+166CFD	- 89 74 24 14	- mov [esp+14],esi
38	PlantsVsZombies.exe+166D01	- E8 8AC0FFFF	- call PlantsVsZombies.exe
39	PlantsVsZombies.exe+166D06	- 2B 74 24 20	- sub esi,[esp+20]
40	PlantsVsZombies.exe+166D0A	- 89 44 24 1C	- mov [esp+1C],eax
41	PlantsVsZombies.exe+166D0E	- 8B C5	- mov eax,ebp
42	PlantsVsZombies.exe+166D10	- 89 B5 C8000000	- mov [ebp+000000C8],esi
43			

可以看到此代码段没有除了函数头的

PlantsVsZombies.exe+166C89 - 8B 6C 24 1, 4 - mov ebp,[esp+14]

没有其他代码给它赋值但是也不一定，也有可能是call进某个函数进行赋值 我们进行验证

mov ebp,[esp+14]时 EBP 225EFB80

而僵尸血量赋值时 EBP 225EFB80

ebp无变化，说明ebp的值是被[esp+14]决定的 而不是堆栈帧的作用了

1	PlantsVsZombies.exe+166C80	- 83 EC 08	- sub esp,08 函数头
2	PlantsVsZombies.exe+166C83	- 8B 44 24 14	- mov eax,[esp+14]
3	PlantsVsZombies.exe+166C87	- 53	- push ebx
4	PlantsVsZombies.exe+166C88	- 55	- push ebp
5	PlantsVsZombies.exe+166C89	- 8B 6C 24 14	- mov ebp,[esp+14]

由上可以看出，ebp正好是call进入这个函数前的esp指向的值

然后我们就要寻找调用这个函数的call指令 这里可能由于CE的问题F8直接跑飞，所以 shift+F8（F8到返回）来到调用函数

PlantsVsZombies:74 09	je	PlantsVsZombies.exe+16720D	
PlantsVsZombies:8B C5	mov	eax,ebp	
PlantsVsZombies:8B CE	mov	ecx,esi	
PlantsVsZombies:E8 23F6FFFF	call	PlantsVsZombies.exe+166830	
PlantsVsZombies:85 C0	test	eax,ecx	
PlantsVsZombies:7E 08	jle	PlantsVsZombies.exe+167219	
PlantsVsZombies:53	push	ebx	
PlantsVsZombies:50	push	eax	
PlantsVsZombies:56	push	esi	
PlantsVsZombies:E8 67FAFFFF	call	PlantsVsZombies.exe+166C80	
>>PlantsVsZombies:5F	pop	edi	
PlantsVsZombies:5D	pop	ebp	
PlantsVsZombies:5B	pop	ebx	
PlantsVsZombies:59	pop	ecx	
PlantsVsZombies:C2 0400	ret	0004	4
PlantsVsZombies:83 F8 05	cmp	eax,05	5

来到这里 这里才是调用僵尸受伤函数的call

ESP 0019F94C call之前 225efb80 和ebp相同

而这个值是怎么来的 是push esi来的

所以 esi寄存器 是我们接下来要跟踪的一个寄存器

此层函数调用没有esi的赋值 假设是在上一层call

1	PlantsVsZombies.exe+92FA0	- 83 EC 14	- sub esp,14
2	PlantsVsZombies.exe+92FA3	- 53	- push ebx
3	PlantsVsZombies.exe+92FA4	- 55	- push ebp
4	PlantsVsZombies.exe+92FA5	- 56	- push esi
5	PlantsVsZombies.exe+92FA6	- 57	- push edi
6	PlantsVsZombies.exe+92FA7	- 8B F0	- mov esi,ecx //esi为ecx赋值
7	PlantsVsZombies.exe+92FA9	- 8B F9	- mov edi,ecx
8	PlantsVsZombies.exe+92FAB	- 56	- push esi
9	PlantsVsZombies.exe+92FAC	- 8B C7	- mov eax,edi
10	PlantsVsZombies.exe+92FAE	- E8 7DFEFFFF	- call PlantsVsZombies.exe+166C80
11	PlantsVsZombies.exe+92FB3	- 8B 47 5C	- mov eax,[edi+5C]
12	PlantsVsZombies.exe+92FB6	- 83 F8 06	- cmp eax,06
13	PlantsVsZombies.exe+92FB9	- 75 21	- jne PlantsVsZombies.exe+92FBB
14	PlantsVsZombies.exe+92FBB	- 85 F6	- test esi,esi
15	PlantsVsZombies.exe+92FBD	- 74 1D	- je PlantsVsZombies.exe+92FBE
16	PlantsVsZombies.exe+92FBF	- 8B 4E 24	- mov ecx,[esi+24]
17	PlantsVsZombies.exe+92FC2	- 83 F9 16	- cmp ecx,16
18	PlantsVsZombies.exe+92FC5	- 74 3E	- je PlantsVsZombies.exe+92FCE
19	PlantsVsZombies.exe+92FC7	- 83 F9 0C	- cmp ecx,0C
20	PlantsVsZombies.exe+92FCA	- 74 39	- je PlantsVsZombies.exe+92FCE
21	PlantsVsZombies.exe+92FCC	- 8B 8E D8000000	- mov ecx,[esi+000000D8]
22	PlantsVsZombies.exe+92FD2	- 83 F9 01	- cmp ecx,01
23	PlantsVsZombies.exe+92FD5	- 74 2E	- je PlantsVsZombies.exe+92FDE
24	PlantsVsZombies.exe+92FD7	- 83 F9 03	- cmp ecx,03

25	PlantsVsZombies.exe+92FDA - 74 29	- je PlantsVsZombies.exe+92
26	PlantsVsZombies.exe+92FDC - 83 F8 03	- cmp eax,03
27	PlantsVsZombies.exe+92FDF - 74 0A	- je PlantsVsZombies.exe+92
28	PlantsVsZombies.exe+92FE1 - 83 F8 05	- cmp eax,05
29	PlantsVsZombies.exe+92FE4 - 74 05	- je PlantsVsZombies.exe+92
30	PlantsVsZombies.exe+92FE6 - 83 F8 06	- cmp eax,06
31	PlantsVsZombies.exe+92FE9 - 75 1A	- jne PlantsVsZombies.exe+92
32	PlantsVsZombies.exe+92FEB - 83 F8 06	- cmp eax,06
33	PlantsVsZombies.exe+92FEE - 75 0B	- jne PlantsVsZombies.exe+92
34	PlantsVsZombies.exe+92FF0 - 85 F6	- test esi,esi
35	PlantsVsZombies.exe+92FF2 - 74 07	- je PlantsVsZombies.exe+92
36	PlantsVsZombies.exe+92FF4 - 8B C6	- mov eax,esi
37	PlantsVsZombies.exe+92FF6 - E8 C5550D00	- call PlantsVsZombies.exe+
38	PlantsVsZombies.exe+92FFB - 56	- push esi
39	PlantsVsZombies.exe+92FFC - 8B C7	- mov eax,edi
40	PlantsVsZombies.exe+92FFE - E8 BDF4FFFF	- call PlantsVsZombies.exe+
41	PlantsVsZombies.exe+93003 - EB 1D	- jmp PlantsVsZombies.exe+
42	PlantsVsZombies.exe+93005 - 85 F6	- test esi,esi
43	PlantsVsZombies.exe+93007 - 74 19	- je PlantsVsZombies.exe+92
44	PlantsVsZombies.exe+93009 - 8B C7	- mov eax,edi
45	PlantsVsZombies.exe+9300B - E8 200C0000	- call PlantsVsZombies.exe+
46	PlantsVsZombies.exe+93010 - 8B D8	- mov ebx,eax
47	PlantsVsZombies.exe+93012 - 8B C6	- mov eax,esi
48	PlantsVsZombies.exe+93014 - E8 47F3FFFF	- call PlantsVsZombies.exe+
49	PlantsVsZombies.exe+93019 - 8B 4B 08	- mov ecx,[ebx+08]
50	PlantsVsZombies.exe+9301C - 51	- push ecx
51	PlantsVsZombies.exe+9301D - E8 3E410D00	- call PlantsVsZombies.exe+
52	PlantsVsZombies.exe+93022 - D9 47 30	- fld dword ptr [edi+30]
53		

所以现在的数​​据流是：[ebp+c8]->ebp->esi->[esi+c8]->[eax+c8]

但是这个调用无法进一步进行跟踪了，这个函数应该是修改返回地址之后 jmp到函数头的，没有办法

上x32dbg日志追踪 将eax == 找到的当前僵尸地址 的eip打印出来 看一看

发现这个函数

00435170	57	push edi	关键函数点
00435171	BF 0000FFFF	mov edi,FFFF0000	
00435176	8B06	mov eax,dword ptr ds:[esi]	
00435178	85C0	test eax,eax	
0043517A	75 08	jne plantsvszombies.435184	
0043517C	8B82 A8000000	mov eax,dword ptr ds:[edx+A8]	edx+A8: "H:\n!\n"
00435182	EB 05	jmp plantsvszombies.435189	
00435184	05 68010000	add eax,168	
00435189	8B8A AC000000	mov ecx,dword ptr ds:[edx+AC]	
0043518F	69C9 68010000	imul ecx,ecx,168	
00435195	038A A8000000	add ecx,dword ptr ds:[edx+A8]	edx+A8: "H:\n!\n"
00435198	3BC1	cmp eax,ecx	
0043519D	73 12	jae plantsvszombies.435181	
0043519F	90	nop	
004351A0	8588 64010000	test dword ptr ds:[eax+164],edi	
004351A6	75 13	jne plantsvszombies.435188	
004351A8	05 68010000	add eax,168	
004351AD	3BC1	cmp eax,ecx	
004351AF	72 EF	jB plantsvszombies.4351A0	
004351B1	C706 FFFFFFFF	mov dword ptr ds:[esi],FFFFFFFF	
004351B7	32C0	xor al,al	
004351B9	5F	pop edi	
004351BA	C3	ret	
004351B8	8906	mov dword ptr ds:[esi],eax	
004351BD	8088 EC000000 00	cmp byte ptr ds:[eax+EC],0	
004351C4	75 80	jne plantsvszombies.435176	
004351C6	80 01	mov al,1	
004351C8	5F	pop edi	
004351C9	C3	ret	

这个地方将esi指向地址存储的值取出来给eax 这里就是僵尸地址 在分析这个函数 是由edx+a8 中的数据进行处理之后 赋值给esi的

每次取出来168 在进行判断，我猜测edx+a8存储的就是僵尸数组首地址，然后不断轮询，将需要处理的僵尸数组元素地址放入[esi]，然后对esi中存储的数据进行处理

经过实验发现，的确是这样

从而 僵尸数组首地址是[edx+a8] ,数组元素大小为168

数组元素偏移为c8的是僵尸的血量

僵尸数组一共10个 从最后开始生成僵尸 僵尸血量和路障不是同一个

c8偏移为血量 cc偏移为最高血量 血量降为50则死亡

d0偏移为护甲 dd偏移为最大护甲

秒杀僵尸：

秒杀僵尸可以是创建一个线程 对僵尸数组进行不停的遍历 修改掉僵尸血量

而经过我们的推测

00435170	57	push edi	关键函数点
00435171	BF 0000FFFF	mov edi,FFFF0000	
00435176	8B06	mov eax,dword ptr ds:[esi]	
00435178	85C0	test eax,eax	
0043517A	75 08	jne plantsvszombies.435184	
0043517C	8B82 A8000000	mov eax,dword ptr ds:[edx+A8]	
00435182	EB 05	jmp plantsvszombies.435189	
00435184	05 68010000	add eax,168	
00435189	8B8A AC000000	mov ecx,dword ptr ds:[edx+AC]	
0043518F	69C9 68010000	imul ecx,ecx,168	
00435195	038A A8000000	add ecx,dword ptr ds:[edx+A8]	

是僵尸初始化或者状态改变所经过的函数调用

所以我们可以对这个函数进行inline HOOK

找到改变的僵尸的地址，

0043518A	C3	ret
0043518B	8906	mov dword ptr ds:[esi],eax

inline Hook 调用这个函数 即可实现僵尸一击必杀

```

1 void __declspec(naked) f(void)
2 {
3     __asm
4     {
5         push ebp;
6         mov ebp, esp;
7         sub esp, 0x10;
8     }
9     PDWORD pdwJiangShi;
10    __asm
11    {
12        mov pdwJiangShi, eax
13    }
14
15    __asm
16    {
17        mov bl,g_bIsMiaosha //判断是否进行秒杀
18        test bl,bl
19        jne Func1
20        leave;
21        ret;
22    }
23
24 Func1:
25     if(*PDWORD((DWORD)pdwJiangShi + JS_XUELIANG_OFFSET) == 270) //还没有被削弱
26         *PDWORD((DWORD)pdwJiangShi + JS_XUELIANG_OFFSET) = 30; //血量不能是0
27     //不同僵尸 有不同的的护甲 需要区别处理
28     *PDWORD((DWORD)pdwJiangShi + JS_HUJIA1_OFFSET) = 0;
29     *PDWORD((DWORD)pdwJiangShi + JS_HUJIA2_OFFSET) = 0;
30
31    __asm
32    {
33        leave;
34        ret;
35    }
36 }
37

```

需要注意的是，这个点是游戏开始时频繁调用的函数点 所以HOOK的时候和卸载HOOK都必须找好时间点 不然游戏会崩溃