## 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


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  看详细汇编

```
PlantsVsZombies.exe+166C80 - 83 EC 08              - sub esp,08      函数头
PlantsVsZombies.exe+166C83 - 8B 44 24 14           - mov eax,[esp+14]
PlantsVsZombies.exe+166C87 - 53                    - push ebx
PlantsVsZombies.exe+166C88 - 55                    - push ebp
PlantsVsZombies.exe+166C89 - 8B 6C 24 14           - mov ebp,[esp+14]
PlantsVsZombies.exe+166C8D - 56                    - push esi
PlantsVsZombies.exe+166C8E - 57                    - push edi
PlantsVsZombies.exe+166C8F - 8B F8                 - mov edi,eax
PlantsVsZombies.exe+166C91 - 83 E7 08              - and edi,08
PlantsVsZombies.exe+166C94 - 89 7C 24 10           - mov [esp+10],edi
PlantsVsZombies.exe+166C98 - 75 07                 - jne PlantsVsZombies.exe+
PlantsVsZombies.exe+166C9A - C7 45 54 19000000     - mov [ebp+54],00000019
PlantsVsZombies.exe+166CA1 - A8 04                 - test al,04
PlantsVsZombies.exe+166CA3 - 74 50                 - je PlantsVsZombies.exe+1
PlantsVsZombies.exe+166CA5 - 8B F5                 - mov esi,ebp
PlantsVsZombies.exe+166CA7 - E8 84060000           - call PlantsVsZombies.exe
PlantsVsZombies.exe+166CAC - 84 C0                 - test al,al
PlantsVsZombies.exe+166CAE - 74 45                 - je PlantsVsZombies.exe+1
PlantsVsZombies.exe+166CB0 - 83 BD AC000000 00     - cmp dword ptr [ebp+00000
PlantsVsZombies.exe+166CB7 - 75 1D                 - jne PlantsVsZombies.exe+
PlantsVsZombies.exe+166CB9 - 8B 45 00              - mov eax,[ebp+00]
PlantsVsZombies.exe+166CBC - 80 B8 E9090000 00     - cmp byte ptr [eax+000009
PlantsVsZombies.exe+166CC3 - 75 11                 - jne PlantsVsZombies.exe+
PlantsVsZombies.exe+166CC5 - 8B 80 88080000        - mov eax,[eax+00000888]
PlantsVsZombies.exe+166CCB - 50                    - push eax
PlantsVsZombies.exe+166CCC - BE 12000000           - mov esi,00000012
PlantsVsZombies.exe+166CD1 - E8 4AB9FDFF           - call PlantsVsZombies.exe
PlantsVsZombies.exe+166CD6 - 8B 85 AC000000        - mov eax,[ebp+000000AC]
PlantsVsZombies.exe+166CDC - 3D E8030000           - cmp eax,000003E8
PlantsVsZombies.exe+166CE1 - 7D 05                 - jnl PlantsVsZombies.exe+
```

```
PlantsVsZombies.exe+166CE3 - B8 E8030000           - mov eax,000003E8
PlantsVsZombies.exe+166CE8 - 8B F5                 - mov esi,ebp
PlantsVsZombies.exe+166CEA - 89 85 AC000000        - mov [ebp+000000AC],eax
PlantsVsZombies.exe+166CF0 - E8 BBDAFFFF           - call PlantsVsZombies.exe
PlantsVsZombies.exe+166CF5 - 8B B5 C8000000        - mov esi,[ebp+000000C8]
PlantsVsZombies.exe+166CFB - 8B C5                 - mov eax,ebp
PlantsVsZombies.exe+166CFD - 89 74 24 14           - mov [esp+14],esi
PlantsVsZombies.exe+166D01 - E8 8AC0FFFF           - call PlantsVsZombies.exe
PlantsVsZombies.exe+166D06 - 2B 74 24 20           - sub esi,[esp+20]
PlantsVsZombies.exe+166D0A - 89 44 24 1C           - mov [esp+1C],eax
PlantsVsZombies.exe+166D0E - 8B C5                 - mov eax,ebp
PlantsVsZombies.exe+166D10 - 89 B5 C8000000        - mov [ebp+000000C8],esi

```

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`

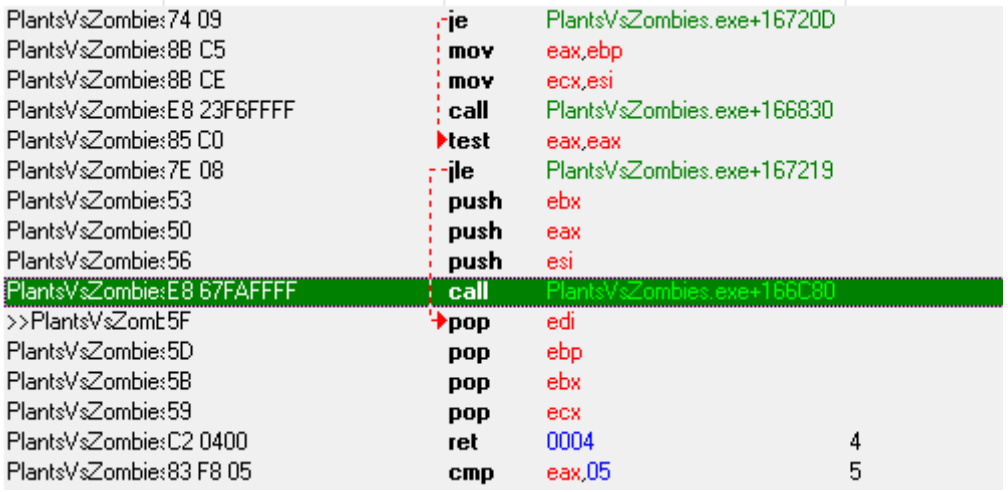
而僵尸血量赋值时 `EBP 225EFB80`

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

```
PlantsVsZombies.exe+166C80 - 83 EC 08              - sub esp,08     函数头
PlantsVsZombies.exe+166C83 - 8B 44 24 14           - mov eax,[esp+14]
PlantsVsZombies.exe+166C87 - 53                    - push ebx
PlantsVsZombies.exe+166C88 - 55                    - push ebp
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




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

```
PlantsVsZombies.exe+92FA0 - 83 EC 14              - sub esp,14
PlantsVsZombies.exe+92FA3 - 53                    - push ebx
PlantsVsZombies.exe+92FA4 - 55                    - push ebp
PlantsVsZombies.exe+92FA5 - 56                    - push esi
PlantsVsZombies.exe+92FA6 - 57                    - push edi
PlantsVsZombies.exe+92FA7 - 8B F0                 - mov esi,eax  //esi为eax赋
PlantsVsZombies.exe+92FA9 - 8B F9                 - mov edi,ecx
PlantsVsZombies.exe+92FAB - 56                    - push esi
PlantsVsZombies.exe+92FAC - 8B C7                 - mov eax,edi
PlantsVsZombies.exe+92FAE - E8 7DFEFFFF           - call PlantsVsZombies.exe+
PlantsVsZombies.exe+92FB3 - 8B 47 5C              - mov eax,[edi+5C]
PlantsVsZombies.exe+92FB6 - 83 F8 06              - cmp eax,06
PlantsVsZombies.exe+92FB9 - 75 21                 - jne PlantsVsZombies.exe+9
PlantsVsZombies.exe+92FBB - 85 F6                 - test esi,esi
PlantsVsZombies.exe+92FBD - 74 1D                 - je PlantsVsZombies.exe+92
PlantsVsZombies.exe+92FBF - 8B 4E 24              - mov ecx,[esi+24]
PlantsVsZombies.exe+92FC2 - 83 F9 16              - cmp ecx,16
PlantsVsZombies.exe+92FC5 - 74 3E                 - je PlantsVsZombies.exe+93
PlantsVsZombies.exe+92FC7 - 83 F9 0C              - cmp ecx,0C
PlantsVsZombies.exe+92FCA - 74 39                 - je PlantsVsZombies.exe+93
PlantsVsZombies.exe+92FCC - 8B 8E D8000000        - mov ecx,[esi+000000D8]
PlantsVsZombies.exe+92FD2 - 83 F9 01              - cmp ecx,01
PlantsVsZombies.exe+92FD5 - 74 2E                 - je PlantsVsZombies.exe+93
PlantsVsZombies.exe+92FD7 - 83 F9 03              - cmp ecx,03
```

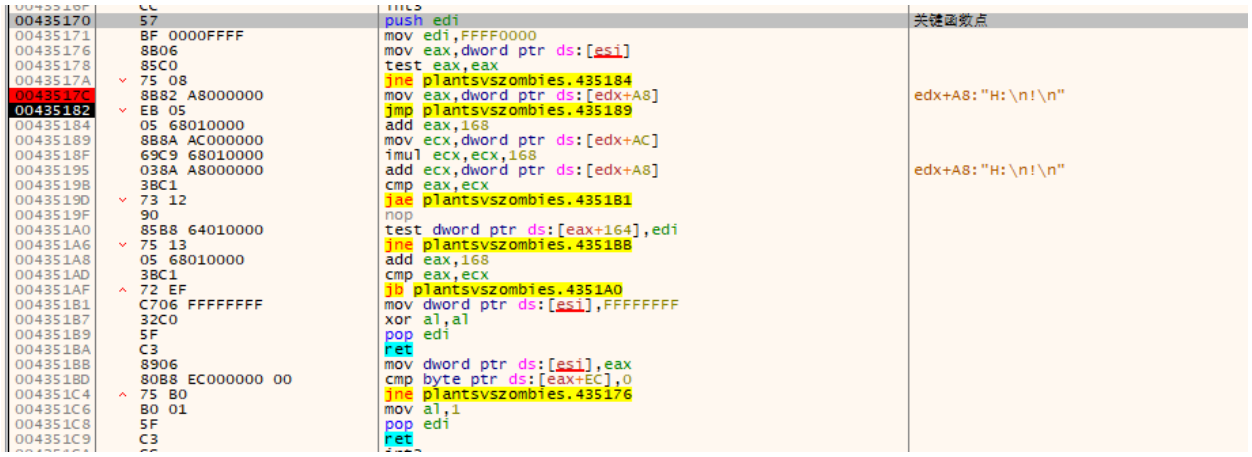
```
PlantsVsZombies.exe+92FDA - 74 29                 - je PlantsVsZombies.exe+93
PlantsVsZombies.exe+92FDC - 83 F8 03              - cmp eax,03
PlantsVsZombies.exe+92FDF - 74 0A                 - je PlantsVsZombies.exe+92
PlantsVsZombies.exe+92FE1 - 83 F8 05              - cmp eax,05
PlantsVsZombies.exe+92FE4 - 74 05                 - je PlantsVsZombies.exe+92
PlantsVsZombies.exe+92FE6 - 83 F8 06              - cmp eax,06
PlantsVsZombies.exe+92FE9 - 75 1A                 - jne PlantsVsZombies.exe+9
PlantsVsZombies.exe+92FEB - 83 F8 06              - cmp eax,06
PlantsVsZombies.exe+92FEE - 75 0B                 - jne PlantsVsZombies.exe+9
PlantsVsZombies.exe+92FF0 - 85 F6                 - test esi,esi
PlantsVsZombies.exe+92FF2 - 74 07                 - je PlantsVsZombies.exe+92
PlantsVsZombies.exe+92FF4 - 8B C6                 - mov eax,esi
PlantsVsZombies.exe+92FF6 - E8 C5550D00           - call PlantsVsZombies.exe+
PlantsVsZombies.exe+92FFB - 56                    - push esi
PlantsVsZombies.exe+92FFC - 8B C7                 - mov eax,edi
PlantsVsZombies.exe+92FFE - E8 BDF4FFFF           - call PlantsVsZombies.exe+
PlantsVsZombies.exe+93003 - EB 1D                 - jmp PlantsVsZombies.exe+9
PlantsVsZombies.exe+93005 - 85 F6                 - test esi,esi
PlantsVsZombies.exe+93007 - 74 19                 - je PlantsVsZombies.exe+93
PlantsVsZombies.exe+93009 - 8B C7                 - mov eax,edi
PlantsVsZombies.exe+9300B - E8 200C0000           - call PlantsVsZombies.exe+
PlantsVsZombies.exe+93010 - 8B D8                 - mov ebx,eax
PlantsVsZombies.exe+93012 - 8B C6                 - mov eax,esi
PlantsVsZombies.exe+93014 - E8 47F3FFFF           - call PlantsVsZombies.exe+
PlantsVsZombies.exe+93019 - 8B 4B 08              - mov ecx,[ebx+08]
PlantsVsZombies.exe+9301C - 51                    - push ecx
PlantsVsZombies.exe+9301D - E8 3E410D00           - call PlantsVsZombies.exe+
PlantsVsZombies.exe+93022 - D9 47 30              - fld dword ptr [edi+30]

```

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




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

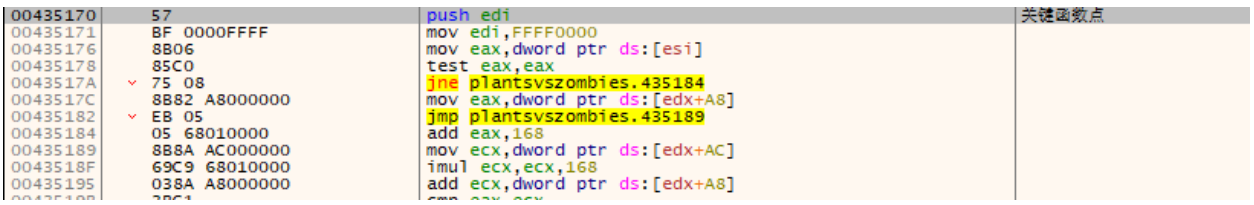
c8偏移为血量 cc偏移为最高血量 血量降为50则死亡

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




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，


inline Hook 调用这个函数 即可实现僵尸一击必杀

```
void __declspec(naked) f(void)
{
    __asm
    {
        push ebp;
        mov ebp, esp;
        sub esp, 0x10;
    }
    PDWORD pdwJiangShi;
    _asm
    {
        mov pdwJiangShi, eax
    }

    __asm
    {
        mov bl,g_bIsMiaosha //判断是否进行秒杀
        test bl,bl
        jne Func1
        leave;
        ret;
    }

Func1:
    if(*PDWORD((DWORD)pdwJiangShi + JS_XUELIANG_OFFSET) == 270) //还没有被削弱
        *PDWORD((DWORD)pdwJiangShi + JS_XUELIANG_OFFSET) = 30; //血量不能是0
    //不同僵尸 有不同的的护甲  需要区别处理
    *PDWORD((DWORD)pdwJiangShi + JS_HUJIA1_OFFSET) = 0;
    *PDWORD((DWORD)pdwJiangShi + JS_HUJIA2_OFFSET) = 0;

    __asm
    {
        leave;
        ret;
    }
}

```

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