第3章作业

姓名：袁昊男 学号：2018091618008

1、简述软件逆向的作用。

**答：**将可执行程序反汇编，通过分析反汇编得到的汇编代码来理解软件功能、逆向分析原始软件的思路，然后使用高级语言重新描述这段代码，这个过程叫做逆向工程。软件逆向作用大致分为以下几类：开发的问题、软件的问题、软件的分析。通过逆向分析可以找出软件测试中没有发现的异常；或是程序能够正常执行，但存在潜在的缓冲区溢出漏洞。可以对软件本身的执行流程进行屏蔽或更改，如禁止软件恶意弹窗、绕过软件版权验证等。还可以对携带病毒的软件进行进一步分析，试探其运行流程，甚至反向分析出软件源代码。

2、系统栈在程序执行中的作用有哪些？

**答：**系统栈（也叫核心栈、内核栈）是内存中属于操作系统空间的一块区域，其主要用途为：

(1) 保存中断现场。对于嵌套中断，被中断程序的现场信息依次压入系统栈，中断返回时逆序弹出；

(2) 保存操作系统子程序间相互调用的参数、返回值、返回点以及子程序（函数）的局部变量。

3、以一个EXE（C程序）反编译代码为例，说明参数的传递和函数执行过程。（注：需按反编译结果给出虚拟地址）

**答：**

|  |  |
| --- | --- |
| **C语言源程序** | **反汇编结果** |
| int Add(int x,int y);  int main()  {  int a=5, b=6;  Add(a,b);  return 0;  }  int Add(int x,int y)  {  return (x+y);  } | 00401000 push 6  00401002 push 5  00401004 call 00401010 ;调用Add()函数  00401009 add esp, 8  0040100C xor eax, eax  0040100E retn  0040100F nop  ……  00401010 push ebp  00401011 mov ebp, esp  00401013 sub esp, 8  00401016 mov eax, dword ptr[ebp + 0C]  0040101A mov ecx, dword ptr[ebp + 08]  0040101E add eax, ecx  00401020 mov esp, ebp  00401022 pop ebp  00401023 retn |

main()函数中调用了Add()子函数，有两个参数a、b。按stdcall调用时，首先从右往左将两个参数压栈（00401000、00401002语句）；然后执行call指令调用Add()函数，此时call指令将main函数返回地址EIP压栈；进入Add()函数中，保存旧栈底并设置新的栈底（00401010、00401011语句），并抬高ESP指针，设置新的栈空间（00401013语句）；通过EBP指针寻址两个参数（00401016、0040101A语句）并将其相加；最后平衡栈、恢复main()函数栈底（0040101E、00401020语句）；最后返回main()函数，此时执行pop eip指令，将栈定内容（main()函数返回地址）pop到EIP中，即可返回main()函数。

4、子过程自身的局部变量可以表示为[ebp+Y]形式，此处Y为正值还是负值，为什么？

**答：**Y为负值。子过程自身的局部变量如果要通过[ebp+Y]的形式进行寻址，则说明该变量存储在子过程调用栈中，此时EBP指向子过程调用栈的栈底，因此Y为负值时指向的才是子过程调用栈中的某局部变量。

5、考虑到内存对齐，计算并显示回答如下数据结构实际所占内存单元大小，并在图中标示出来数据的实际存储情况。

|  |
| --- |
| 0x1111 |
| 0x2222 |
| 0x2222 |
| 0x3333 |
| 0x3333 |
|  |
|  |

struct AAA{ 从此处开始填（低端地址）●

short a;

int b;

int c;

} bbb;

bbb.a = 0x1111;

bbb.b = 0x22222222;

bbb.c = 0x33333333;

**答：**结构体在内存中的对其规则如下：

(1) 结构体中元素是按照定义顺序一个一个放到内存中去的，但并不是紧密排列的。从结构体存储的首地址开始，每一个元素放置到内存中时，它都会认为内存是以它自己的大小来划分的，因此元素放置的位置一定会在自己**宽度的整数倍**上开始（以结构体变量首地址为0计算）。

(2) 在经过(1)分析后，检查计算出的存储单元是否为所有元素中最宽的元素的长度的整数倍，是，则结束；若不是，则补齐为它的整数倍。

本题中：一个内存单元占4字节。short a占2字节，放入第一个内存单元中；int b和int c分别占4字节，放入第二、三个内存单元中。此时共占用3个内存单元，共12字节。结构体中最宽的元素是int，占4字节。因此按(2)原则，应该补齐为4字节的最小整数倍，共计12字节。