SEHAttack

一．SEH

S.E.H即异常处理结构（Structure Exception Handler）,它是Windows异常处理机制所用的重要数据结构，每个SEH包含两个DWORD指针

|  |
| --- |
| DWORD:Next SEH(下一个SEH指针) |
| DWORD:Exception Handler(异常处理函数) |

(1)SEH结构体存放在系统栈中

(2)当线程初始化时，系统会自动向栈中安装一个SEH，作为线程默认的异常处理

(3)如果程序源代码使用了\_\_try{} \_\_except(){}或者\_\_Assert宏等异常处理机制，编译器会向当前函数栈中安装一个SEH

(4)栈中存在多个SEH，通过链表从栈底(高地址)指向栈顶(低地址),当异常出现时，会找最近的SEH处理,当处理失败时，将顺着SEH链表依次尝试其他的异常处理函数，如果都不适合，将调用系统默认的处理函数

二．在栈溢出中使用SEH

char shellcode[]="\x90\x90\x90\x90"

"\x90\x90\x90\x90"

"\x90\x90\x90\x90";

DWORD MyException(void){

printf("error !!!exit the process...");

getchar();

ExitProcess(1);

}

void test(char \*input){

char buf[20];

int zero=0;

\_\_asm int 3

\_\_try{

strcpy(buf,input);

zero=4/zero;

}

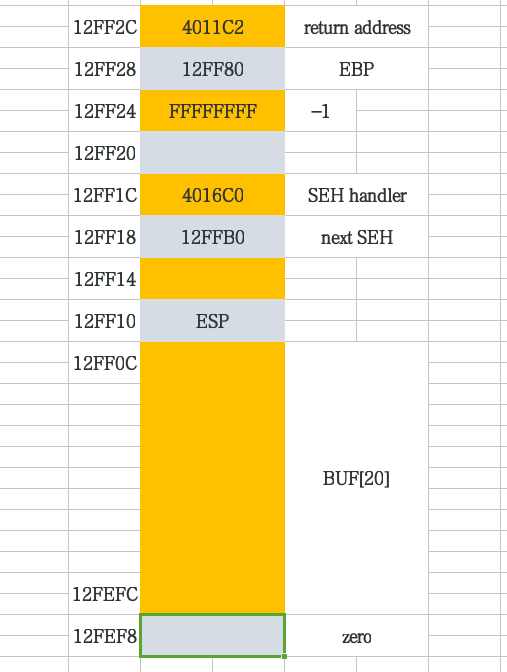
\_\_except (MyException()){}

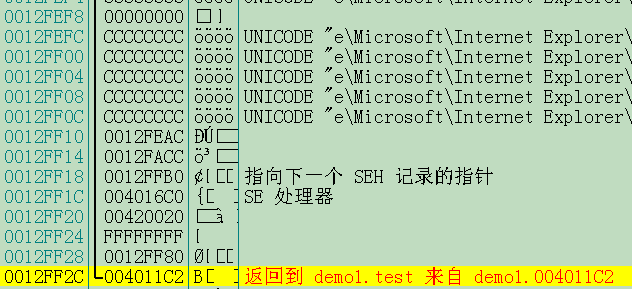
}

main(){

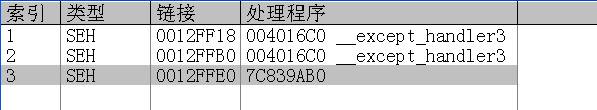
test(shellcode);

}

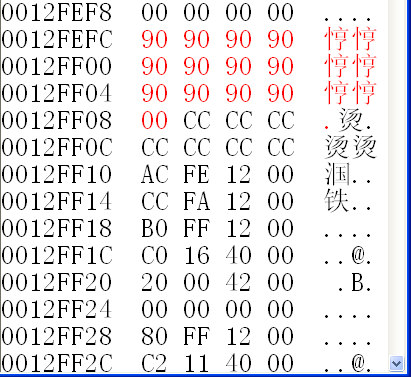




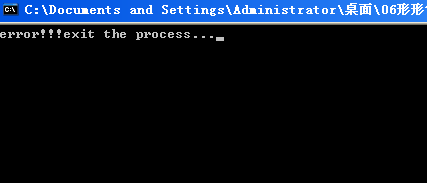
SEH链



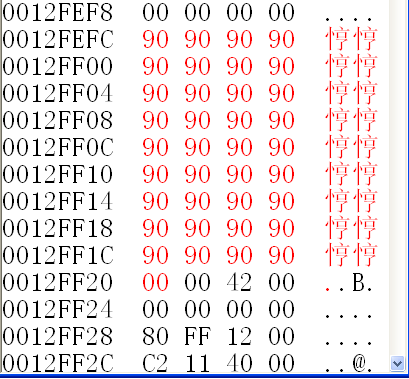
strcpy



zero=4/zero



当shellcode溢出时



zero=4/zero;



三．在堆溢出利用SEH

即利用DWORD SHOOT修改SEH handler