Shellcode

1.简介

在计算机安全中，shellcode是一小段代码，可以用于软件漏洞利用的载荷。被称为“shellcode”是因为它通常启动一个命令终端，攻击者可以通过这个终端控制受害的计算机，但是所有执行类似任务的代码片段都可以称作shellcode。Shellcode通常是以机器码形式编写的。

你可能想要利用已有的标准shellcode，比如来自Shell Storm数据库或由Metasploit的msfvenom工具生成。

二．shellcode实验

1.采用“跳板”定位shellcode

因为动态链接库的装入与卸载，Windows进程的函数栈帧很可能会产生“移位”，即shellcode在内存中的地址会动态变化，则可以将返回地址覆盖为jmp esp指令地址，将shellcode填充在返回地址之后

int verify\_password (char \*password)

{

int authenticated;

char buffer[44];

authenticated=strcmp(password,PASSWORD);

strcpy(buffer,password);

return authenticated;

}

main()

{

int valid\_flag=0;

char password[1024];

FILE \* fp;

LoadLibrary("user32.dll"); if(!(fp=fopen("password.txt","rw+")))

{

exit(0);

}

fscanf(fp,"%s",password);

valid\_flag = verify\_password(password);

if(valid\_flag)

{

printf("incorrect password!\n");

}

else

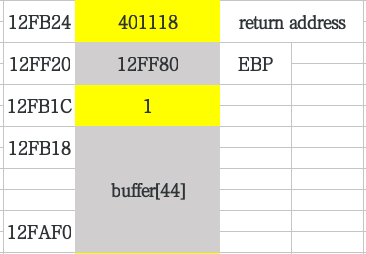
{

printf("Congratulation! You have passed the verification!\n");

}

fclose(fp);

}

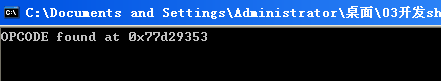


return address:401118

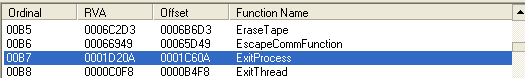
EBP:12FF80

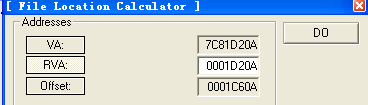
(1)获取JMP ESP（user32.dll）的地址

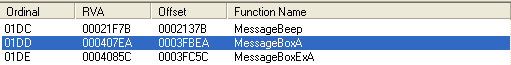
<search_opcode.cpp>

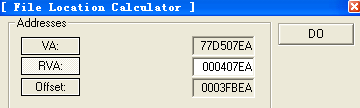


(2)获取messageboxA(0,”str”,”str”,0),exitprocess(0);





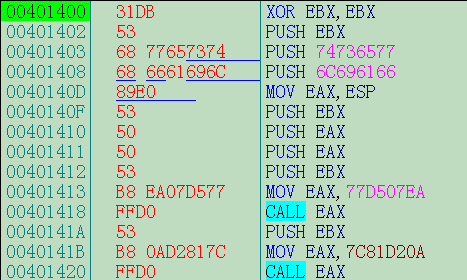


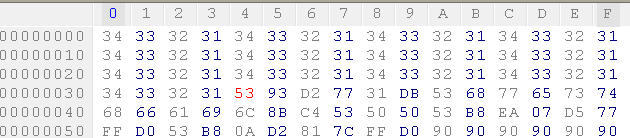


MessageboxA:77D507EA

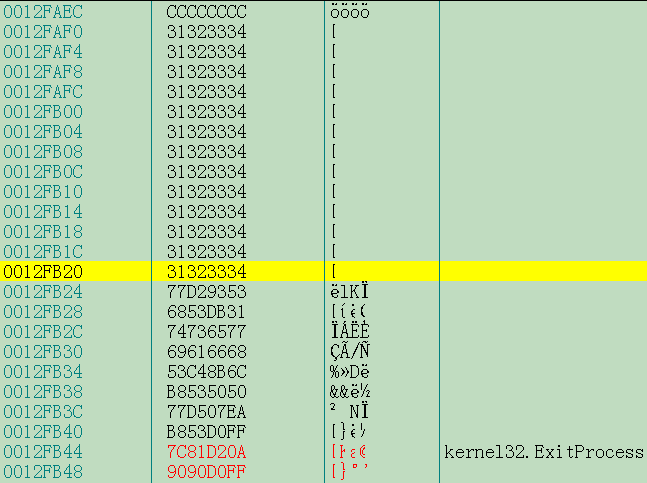
ExitProcess:7C81D20A

(3)shellcode编写





(4)结果



2.其他技术

（1）抬高栈顶：既可以保护shellcode,又可以尽量不破坏高地址栈的数据

（2）不使用跳转指令:有些系统对kernel32.dll文件内容不可读，则可以用shellcode盲射,即使用大量的nops指令

3.shellcode编码

shellcode编码：（1）空字节（NULL）的取值为：0x00。在C/C++代码中，空字节被认为是字符串的结束符。则可以避免NULL字符

（2）.可以逃避IDS的非法字符串检测