

bit2font vulnerabilities discovery

0x1 Introduction

The package **bm2font** accepts graphic input in one of several standard bitmap graphic formats, and converts it to "fonts", each glyph of which covers a tile of the image.












It is a package widely used in text editors, such as textlive, etc.

<https://www.textlive.info/CTAN/graphics/bm2font/>

The latest version 3.0 package can be found at:

<https://ctan.org/tex-archive/graphics/bm2font>

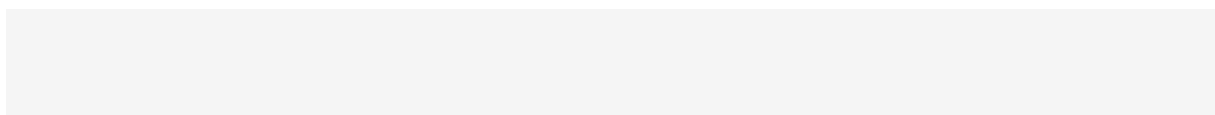
Files

Name	Size	Date	Notes
 Gnu.licence	12 kB	1994-02-25	
 Makefile	926	1994-02-25	
 Makefile.dos	418	1994-02-25	
 README	4 kB	1994-02-25	
 bm2font-win32.exe	75 kB	2000-12-25	
 bm2font.c	111 kB	1994-10-26	
 bm2font.exe	77 kB	1994-04-29	
 loc.p2clib.c	124	1994-02-25	
 manual.zip	482 kB	1993-06-15	
 p2c.h	13 kB	1994-02-25	
 p2clib.c	18 kB	1994-02-25	

0x2 Vul1

0x2.1 Crash scene

After executing bm2font with a parameter length over 256 (no matter such a file exists), bm2font gets crash.



[illegible]

using gdb to debug:
set breakpoint at 0x405CA5 and go to next instruction.

```

0x00007ffffffffffc60 +0x0020: 0x0000000000000001
0x00007ffffffffffc68 +0x0028: 0x0000000000000000
0x00007ffffffffffc70 +0x0030: 0x00007ffffffffffcfe8 → 0x00007ffffffffffd890 → 0x00007ffffffffffa280 → 0x0000001200000036 ("6"? )
0x00007ffffffffffc78 +0x0038: 0x0000000000000000

[0] Id 1, Name: "bm2font", stopped, reason: BREAKPOINT
[0] 0x405ca5 → main()
gef> nl
=====
==82006==ERROR: AddressSanitizer: strcpy-param-overlap: memory ranges [0x00000062d0e0,0x00000062d1e4] and [0x00000062cfe0, 0x00000062d0e0] overlap
#0 0x7ffff6ecc78e (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x6278e)
#1 0x405ca9 in main (/home/test/Desktop/evaluation/bm2font/bm2font/bm2font+0x405ca9)
#2 0x7ffff67b782f in _libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
#3 0x40d1e8 in _start (/home/test/Desktop/evaluation/bm2font/bm2font/bm2font+0x40d1e8)

AddressSanitizer can not describe address in more detail (wild memory access suspected).
AddressSanitizer can not describe address in more detail (wild memory access suspected).
SUMMARY: AddressSanitizer: strcpy-param-overlap ??:0 ??
==82006==ABORTING
[Inferior 1 (process 82006) exited with code 01]

```

0x2.2 Cause of vulnerability

The main function of `bm2font` lacks the check of input parameter length. When the first parameter is a character string that does not start with the symbol '-' and has a length greater than 1, it will eventually be copied to the variable `font`.

```
3733 printf("-z<area of gradation>      (in %, std 70)\n");
3734 printf("-m<width of picture on paper> (in mm)\n");
3735 printf("-n<height of picture on paper> (in mm)\n");
3736 printf("-j<clip off white space>      (y or n, std y)\n");
3737 printf("-k<color: c,m,y or k>         (std k)\n");
3738 /*readln(comment);
3739 if length(comment)=0 then goto 9999;j:=ord(comment[0])+1;
3740 if j<=127 then move(comment[0],mem[prefixseg:$80],j);*/
3741 goto _L9999;
3742 }
3743 /*:110*/
3744 /*111:*/
3745 for (j = 1; j < P_argc; j++) {
3746     strcpy(comment, P_argv[j]);
3747     if (comment[0] != '-') {
3748         strcpy(font, comment);
3749         if (strlen(font) < 1 || strcmp(font, " ")) {
3750             printf("give the name of your bitmap file next time\n");
3751             goto _L9999;
3752         }
3753     } else {
3754         switch (comment[1]) {
3755             case 'h':
3756             case 'H':
3757                 strdelete((Anyptr)comment, 1, 2);
3758                 i = (sscanf(comment, "%lg", &truehres) == 0);
3759                 break;
3760             case 'v':
```

The `font` will be copied to `bmname`. And a crash is triggered when the length of `font` is more than 256.

```

94 Static ebts b2[3000];
95 Static ebts b3[3000];
96 Static ebts b4[3000];
97 Static long linepos_, posbit;
98 Static halfk dbuf;
99 Static boolean clipon, zeroline, zerorow, nameok, fok;
100 Static Char cmd[256];
101 Static Char bname[256];
102 Static Char font[256];
103 Static Char tmpname[256];
104 Static Char fontpre[256];a
105 Static Char fontupc[256];
106 Static ebts curpat[2];

```

0x2.3 Reproduce

Just run cmd

[illegible]

or run `bm2font` with a `bmp` with its path prefix (length of `bmp` file name is 255))

[illegible]

The program directly copies the first received parameter to the variable **comment** and judge the value of the first and the second characters.

```

3745 for (j = 1; j < P_argc; j++) {
3746     strcpy(comment, P_argv[j]);
3747     if (comment[0] != '-') {
3748         strcpy(font, comment);
3749         if (strlen(font) < 1 || !strcmp(font, " ")) {
3750             printf("give the name of your bitmap file next time\n");
3751             goto _L9999;
3752         }
3753     } else {
3754         switch (comment[1]) {
3755 |
3756         case 'h':
3757         case 'H':
3758             strdelete((Anyptr)comment, 1, 2);
3759             i = (sscanf(comment, "%lg", &truehres) == 0);
3760             break;
3761
3762         case 'v':
3763         case 'V':
3764             strdelete((Anyptr)comment, 1, 2);
3765             i = (sscanf(comment, "%lg", &truevres) == 0);
3766             break;
3767

```

When the first character is '-' and the second character is 'f', it copies the variable **comment** to the **aliasname** after deleting the prefix '-f'.

```

3836 case 'f':
3837 case 'F':
3838     strdelete((Anyptr)comment, 1, 2);
3839     strcpy(aliasname, comment);
3840     aliasused = true;
3841     break;
3842
3843 case 's':
3844 case 'S':
3845     strdelete((Anyptr)comment, 1, 2);
3846     ledprinter = (!strcmp(comment, "y") || !strcmp(comment, "Y"));
3847     break;
----
```

If a suffix length of '-f' of 256 , it will trigger a stack overflow.

```

132 Static FILE *texfile;
133 Static FILE *bitmap, *pixmap, *tmpfil;
134 Static short filestat;
135 Static unsigned short gi;
136 Static boolean invert;
137 Static unsigned short cv, dv, lv;
138 Static Char comment[256];
139 Static Char aliasname[256];
140 Static boolean aliasused;
141 Static double mapdiv;
142 Static long pkwidth, pkheight;
143 Static boolean nowwhite, halfinch;
144 Static pxlstr nextmemfree;

```

0x3.3 Reproduce

Just run cmd

[illegible]

Note

The causes of above vulnerabilities are orthogonal, they caused stack overflow in different locations.