



Published on *Ramz Afzar* (<https://www.rafzar.com>)

[Home](#) > VA010-003 - Adobe Acrobat Font Parsing Integer Overflow Vulnerability

VA010-003 - Adobe Acrobat Font Parsing Integer Overflow Vulnerability

```
# TTF Font Parser
# RamzAfzar Vulnerability Analysis Team

use IO::File;
use warnings;

my $n = 0;
my $buf = "";
my $path = "";
my $fh = IO::File->new();
my $num_args = @ARGV;

if ( $num_args != 1 )
{
    print " \r\n\r\n Usage: \r\n parse.pl fontname.ttf \r\n\r\n";
    exit;
}

$path = $ARGV[0];
print $path;
open $fh, $path or die $!;
binmode $fh;

$n = read $fh, $buf, 12;
if ( $n < 1 )
{
    print "Error in reading file!";
    exit;
}

my ( $version, $number ) = unpack( "Nn", $buf );
print " Font Version = $version, No. of tables = $number\n";
print "\nTABLE\tOFFSET\tLENGTH\n";

for ( my $i = 0 ; $i < $number ; $i++ ) {
    read( $fh, $buf, 16 );
    my ( $table, $offset, $length ) = unpack( "a4x4NN", $buf );
    $table->{$table} = $offset;
    print "$table\t$offset\t$length\n";
}
```

Then I downloaded a new font for my system by searching internet. Finally I started typing some text in Adobe Acrobat Professional with downloaded font. By default Adobe Acrobat Professional 9 embeds used fonts in document. So I didn't changed anything to achieve this goal.

When I saved the document, it was compressed by default, so I used PDFTK to uncompress it:

```
pdftk.exe test.pdf output test_unc.pdf uncompress
```

Then I opened the uncompressed file in Notepad++ and I saw the embedded font data. Till now everything is ok.

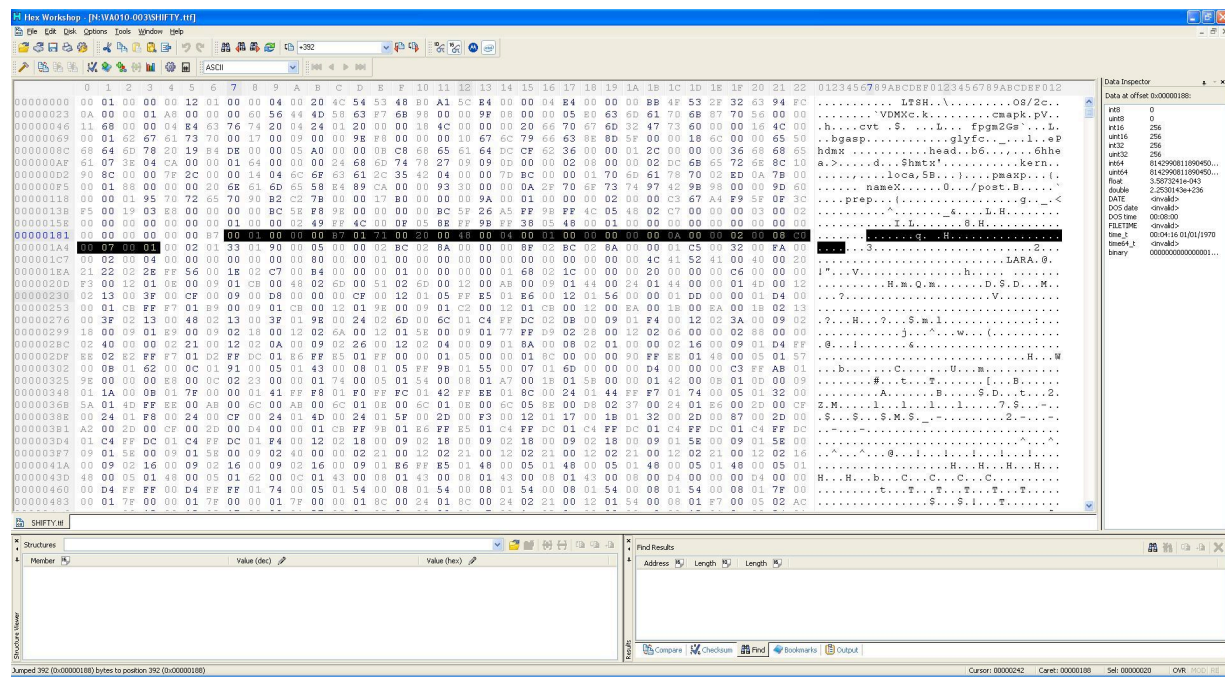
I executed my perl script on the downloaded font, here is results:

```
N:\VA010-003>parse.pl SHIFTY.ttf
N:\Font Stuff>parse.pl SHIFTY.ttf
SHIFTY.ttf Font Version = 65536, No. of tables = 18

TABLE  OFFSET  LENGTH
LTSH   1252    187
OS/2   424     96
VDMX   40712   1504
cmap   4456    1252
cvt    6220     32
fpgm   5708    354
gasp   40696    16
glyf   6252   25936
hdmx   1440   3016
head   300     54
hhea   356     36
hmtx   520    732
kern   32556  5124
loca   32188   368
maxp   392     32
name   37680  2607
post   40288   405
prep   6064    154

N:\VA010-003>
```

Ok, as you may have read, the Integer overflow exists in maxp table, so I go to offset 392 of file and here is what I got:



In Hexadecimal format, here is MAXP Table of my font:

00 01 00 00 00 B7 01 71 00 20 00 48 00 04 00 01 00 00 00 00 00 0A 00 00 02 00 08 C0
00 07 00 01

Here we parse the maxp table using instructions given in this TTF specs:

Type	Name	Description
Fixed	version	0x00010000 (1.0)
uint16	numGlyphs	the number of glyphs in the font
uint16	maxPoints	points in non-compound glyph
uint16	maxContours	contours in non-compound glyph
uint16	maxComponentPoints	points in compound glyph
uint16	maxComponentContours	contours in compound glyph
uint16	maxZones	set to 2
uint16	maxTwilightPoints	points used in Twilight Zone (Z0)
uint16	maxStorage	number of Storage Area locations
uint16	maxFunctionDefs	number of FDEFs
uint16	maxInstructionDefs	number of IDEFs

uint16	maxStackElements	maximum stack depth
uint16	maxSizeOfInstructions	byte count for glyph instructions
uint16	maxComponentElements	number of glyphs referenced at top level
uint16	maxComponentDepth	levels of recursion, set to 0 if font has only simple glyphs

from: <http://developer.apple.com/fonts/TTRefMan/RM06/Chap6maxp.html> [2]

So output of our parsing will be this:

version 0x00010000 (1.0)

numGlyphs 0x00B7

maxPoints 0x0171

maxContours 0x0020

maxComponentPoints 0x0048

maxComponentContours 0x0004

maxZones 0x0001

.....etc.

As document says integer overflow exists in maxComponentPoints, so we have to modify 0x0048 to a large integer value, I modified it to 0xFFFF. Viola! I got crash in Latest version of Adobe Reader and Professional.

I uploaded all files I worked on, including font and PoC file, I tried to add Javascript heap spray in order to get code execution, but it's not that easy to get code execution in this vulnerability, as variable initialization and freeing happens after completion of heap spray and it always looks for an address behind sprayed area of memory.

Here is all files: <https://www.rafzar.com/files/VA010-003.zip> [3]

Any questions? Feel free to ask: va@rafzar.com [4]

Regards

Vulnerability Analysis

Source URL: <https://www.rafzar.com/node/21>

Links:

[1] https://www.rafzar.com/images/maxp_table.jpg

[2] <http://developer.apple.com/fonts/TTRefMan/RM06/Chap6maxp.html>

[3] <https://www.rafzar.com/files/VA010-003.zip>

[4] <mailto:va@rafzar.com>