

# Apple ATSServer bug

the return of the evil charstrings

Anibal Sacco

Matias Eissler

# Freetype < 2.4.2

## Compact font format



```
typedef struct CFF_Decoder_  
{  
    /* ... */  
    FT_Fixed      stack[CFF_MAX_OPERANDS + 1];  
    FT_Fixed*     top;  
    /* ... */  
} CFF_Decoder;  
  
cff_slot_load( CFF_GlyphSlot glyph,  
               CFF_Size      size,  
               FT_UInt       glyph_index,  
               FT_Int32      load_flags )  
{  
    /* ... */  
    CFF_Decoder decoder; // decoder.stack array en el stack  
    /* ... */  
    error = cff_decoder_parse_charstrings( &decoder, charstring, charstring_len );  
    /* ... */  
}  
  
cff_decoder_parse_charstrings( CFF_Decoder* decoder,  
                               FT_Byte*     charstring_base,  
                               FT_ULong     charstring_len )  
{  
    /* ... */  
    decoder->top = decoder->stack;  
    limit = zone->limit = charstring_base + charstring_len; // controlado por usuario  
    /* ... */  
    while(ip < limit)  
    {  
        /* ... */  
        *decoder->top++ = val; // buffer overflow  
    }  
}
```

# Compact Font Format File

0000000	0100	0401	0001	0101	1341	4243	4445	462b	?.???.?????ABCDEF+
0000010	5469	6d65	732d	526f	6d61	6e00	0101	011f	Times-Roman.????
0000020	f81b	00f8	1c02	f81d	03f8	1904	1c6f	000d	???.?????o.?
0000030	fb3c	fb6e	fa7c	fa16	05e9	11b8	f112	0003	?<?n?   ???????.?
0000040	0101	0813	1830	3031	2e30	3037	5469	6d65	??????001.007Time
0000050	7320	526f	6d61	6e54	696d	6573	0000	0002	s RomanTimes...?
0000060	0101	0203	0e0e	7d99	f92a	99fb	7695	f773	??????}??*??v??s
0000070	8b06	f79a	93fc	7c8c	077d	99f8	5695	f75e	??????   ??}??V??^
0000080	9908	fb6e	8cf8	7393	f710	8b09	a70a	df0b	???n??s?????????
0000090	f78e	14							???

start 1    end 2

offSize    start 2

Count

# Mac OS X Leopard



## Thread 1 Crashed:

0	???	0000000000 0 + 0
1	libSystem.B.dylib	0x90995155 <u>_pthread_start</u> + 321
2	libSystem.B.dylib	0x90995012 thread_start + 34

## Thread 0:

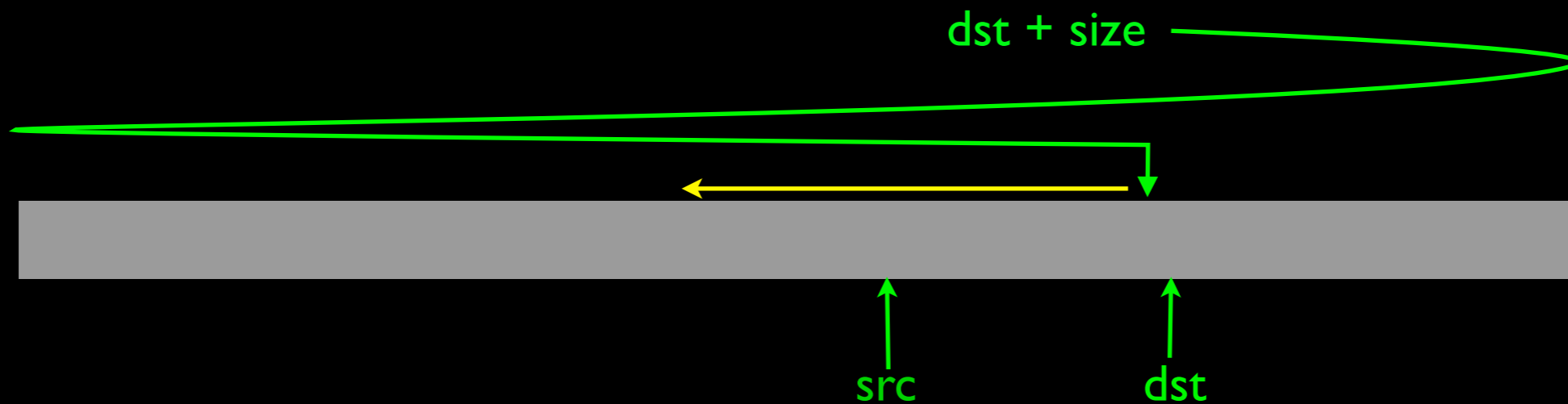
0	libSystem.B.dylib	0xffff0f32 <u>__memcpy</u> + 1938
1	ATSServer	0x00042b11 0x1000 + 269073
2	ATSServer	0x0002e278 0x1000 + 184952
3	ATSServer	0x0002d59e 0x1000 + 181662
4	ATSServer	0x0002d23f 0x1000 + 180799

# Sign mismatch

memcpy(  
dst,  
src,  
0xffffffff)

```
text:000433CD mov     eax, ds:off_2840DD[ebx]
text:000433D3 mov     dword ptr [esp+8], 0
text:000433DB mov     eax, [eax]
text:000433DD mov     [esp+4], eax
text:000433E1 movsx   eax, word ptr [esi+4]
text:000433E5 mov     [esp], eax
text:000433E8 call   sub_7800
text:000433ED mov     edx, [ebp+8]
text:000433F0 mov     edi, eax
text:000433F2 movsx   eax, word ptr [edx+5Eh]
text:000433F6 mov     [esp+0Ch], eax
text:000433FA movsx   eax, word ptr [esi+4] ; Oops! :D
text:000433FE mov     [esp], edi
text:00043401 mov     [esp+8], eax ; Size
text:00043405 mov     eax, [ebp-2Ch]
text:00043408 mov     [esp+4], eax
text:0004340C call   memcpy_wrapper
text:00043411 mov     ecx, [ebp-44h]
text:00043414 mov     [esp], ecx
text:00043417 call   sub_122AD0
text:0004341C mov     ecx, [ebp-40h]
text:0004341F mov     [esp], ecx
text:00043422 call   near ptr sub_122A70
text:00043427 mov     eax, edi
text:00043429 mov     ebx, [ebp-0Ch]
text:0004342C mov     esi, [ebp-8]
text:0004342F mov     edi, [ebp-4]
text:00043432 leave
text:00043433 retn
```

# memcpy va para atras



```
FFFF07A0 memcpy_imp proc near
FFFF07A0
FFFF07A0 dst= dword ptr 8
FFFF07A0 src= dword ptr 0Ch
FFFF07A0 size= dword ptr 10h
FFFF07A0
FFFF07A0 push    ebp
FFFF07A1 mov     ebp, esp
FFFF07A3 push    esi
FFFF07A4 push    edi
FFFF07A5 mov     edi, [ebp+dst]
FFFF07A8 mov     esi, [ebp+src]
FFFF07AB mov     ecx, [ebp+size]
FFFF07AE mov     edx, edi
FFFF07B0 sub     edx, esi
FFFF07B2 cmp     edx, ecx
FFFF07B4 jb     short loc_FFFF07E4 ; if(dst - src > size)
```

# Mac OS X Leopard



## Thread 1 Crashed:

```
0  ???                                0000000000 0 + 0
1  libSystem.B.dylib                 0x90995155  _pthread_start + 321
2  libSystem.B.dylib                 0x90995012  thread_start + 34
```

## Thread 0:

```
0  libSystem.B.dylib                 0xffff0f32  __memcpy + 1938
1  ATSServer                          0x00042b11  0x1000 + 269073
2  ATSServer                          0x0002e278  0x1000 + 184952
3  ATSServer                          0x0002d59e  0x1000 + 181662
4  ATSServer                          0x0002d23f  0x1000 + 180799
```

# Mac OS X Leopard

## Thread 1 Crashed:

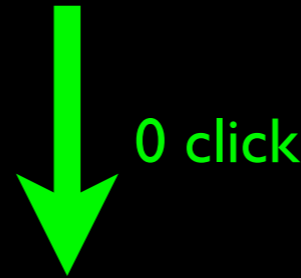
0	???	000000000000 0 + 0
1	libSystem.B.dylib	0x90995155 <b>__pthread_start</b> + 321
2	libSystem.B.dylib	0x90995012 thread_start + 34

# Font spray

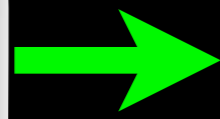
- Tenemos un memcpy de size 0xfffffffffa
- Con ese memcpy alcanzamos el ptr de pthread\_start
- Identificamos el chunk de memoria que va a sobrescribir el ptr.
- Llenamos ese segmento de memoria con datos controlados embebiendo muchas font
- Mission accomplished!

# Attack

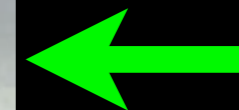
# vectors



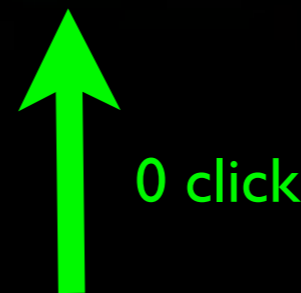
0 click



0 click



1 click



0 click



## . Report Timeline

- ...
- **2010-09-28:** Apple acknowledges the communication informing that this issue will be fixed in the next security update of Mac OS X 10.5, which is tentatively scheduled for the end of October without a firm date of publication.
- **2010-08-31:** Apple asks Core about credit information for the advisory.
- **2010-09-28:** Core acknowledges the communication sending the credit information for this report.
- **2010-10-20:** Core asks Apple for a firm date for the release of this security issue since the initial proposed timeframe of October 18th is due.
- **2010-10-22:** Apple acknowledges the communication informing that the publication date is scheduled to the week of October 25th. Also, Apple notifies that the assigned identifier for this vulnerability is CVE-2010-1797.
- **2010-11-01:** Core asks Apple for a new schedule for the publication, since there was no notice of any Apple security update during the week of October 25th.
- **2010-11-01:** Apple acknowledges the communication informing that the publication date was rescheduled to the middle of the week of November 1st.
- **2010-11-03:** Core informs Apple that the publication of this advisory was scheduled to Monday 8th, taking into account the last communication this is a final publication date. Core also informs that the information about how this vulnerability was found and how it can be exploited will be discussed in a small infosec related local event in Buenos Aires city.
- **2010-11-08:** Core publishes advisory CORE-2010-0825.

that the publication date is scheduled to the week of October 25th.  
Also, Apple notifies that the assigned identifier for this vulnerability is CVE-2010-1797.

- **2010-11-01:** Core asks Apple for a new schedule for the publication, since there was no notice of any Apple security update during the week of October 25th.
- **2010-11-01:** Apple acknowledges the communication informing that the publication date was rescheduled to the middle of the week of November 1st.
- **2010-11-03:** Core informs Apple that the publication of this advisory was scheduled to Monday 8th, taking into account the last communication this is a final publication date. Core also informs that the information about how this vulnerability was found and how it can be exploited will be discussed in a small infosec related local event in Buenos Aires city.
- **2010-11-08:** Core publishes advisory CORE-2010-0825.



**CORE**  
SECURITY TECHNOLOGIES