In-Depth Analysis of Microsoft Office PowerPoint LinkedSlideAtom10 Integer Overflow Vulnerability (CVE-2009-0221)

Table of Contents

Introduction

A vulnerability exists in Microsoft Office PowerPoint in the way it parses a specially crafted Powerpoint presentation, which could allow attackers to execute arbitrary code.

Tested Versions

The vulnerability was analysed on Windows XP SP2 with Microsoft Office PowerPoint 2003 SP3 (Powerpnt.exe version 11.0.8227.0).

Fixed Versions

The vulnerability was addressed with the MS09-017 patch.

Technical Details

A Powerpoint document may embed containers like Handout, MainMaster, Notes or Slides to record data used in the different parts of the presentation. Each of these four containers can contain several atoms, some of them being optional.

An integer overflow vulnerability may be triggered due to insufficient checks performed on the LinkedSlideAtom10 (opcode 12007 or 2EE7h) which is an optional atom belonging to the quoted containers.

The vulnerable code is located in sub_300072AA, reached from sub_3009A622 which parses the different atoms found in a container:

```
.text:301E3A0D loc_301E3A0D:
.text:301E3A0D          cmp     ax, 2EE7h               //LinkedSlideAtom10
.text:301E3A11          jnz     short loc_301E3A59
.text:301E3A13          mov     ecx, [ebp+14h]
.text:301E3A16          push    8
.text:301E3A18          call    sub_3009C88B            //get two parameters
.text:301E3A1D          push    14h
.text:301E3A1F          mov     ebx, eax
.text:301E3A21          call    sub_30004D65            //allocate a 14 bytes long buffer
.text:301E3A26          mov     esi, eax
.text:301E3A28          mov     [ebp-5Ch], esi
.text:301E3A2B          mov     eax, [ebx+4]            //eax = size parameter of LinkedSlideAtom10
.text:301E3A2E          mov     ecx, [ebx]              //ecx = slideIndex parameter
.text:301E3A30          xor     ebx, ebx
.text:301E3A32          push    ebx
.text:301E3A33          push    ebx
.text:301E3A34          push    eax                     //push size
.text:301E3A35          mov     [esi], ecx
.text:301E3A37          push    8                       //this number is used later in a multiplication
.text:301E3A39          lea     ecx, [esi+4]
.text:301E3A3C          mov     dword ptr [ebp-4], 6
.text:301E3A43          call    sub_3001E0B0            //call to a function that calls the vulnerable
                                                        //sub_300072AA function
```

Then, in the vulnerable function:

```
.text:300072AA          push    ebp
.text:300072AB          mov     ebp, esp
.text:300072AD          cmp     [ebp+arg_4], 1          //arg_4 is set to 1
.text:300072B1          push    ebx
.text:300072B2          push    esi
.text:300072B3          push    edi
.text:300072B4          mov     esi, ecx
.text:300072B6          jbe     loc_3001E11D
```

```
…
.text:3001E11D loc_3001E11D:
.text:3001E11D              mov     ebx, [ebp+arg_0]      //ebx = size parameter of LinkedSlideAtom10

.text:3001E120              jmp     loc_300072CC
…
.text:300072CC
.text:300072CC loc_300072CC:
.text:300072CC              mov     esi, [esi+0Ch]        //[esi+0Ch] = 8
.text:300072CF              imul    esi, ebx
.text:300072D2              push    esi
.text:300072D3              call    sub_30004D65          //allocate a new buffer
```

The problem lies in the allocation of this buffer. If ebx is large enough (I.e. when ebx >= 20000000h) the multiplication overflows which results in an undersized buffer allocated.

This buffer is later used to hold data stored in the LinkedShapeAtom10 atoms that normally follow a LinkedSlideAtom10 atom. If enough atoms are provided, a heap overflow occurs leading to arbitrary code execution.

Exploitation

The resulting crash eventually appears in Powerpnt.exe at 0x30021DF6:

```
.text:30021DF2              mov     eax, [ecx]            //[ecx] has been overwritten with data coming
                                                          //from a LinkedShapeAtom10 atom
.text:30021DF4              push    1
.text:30021DF6              call    dword ptr [eax+8]     //eax controlled. It usually equals the linkedIndex
                                                          //parameter of a LinkedShapeAtom10 atom
```

The provided exploit uses a LinkedSlideAtom10 with a size set to 20000001h. Then a few LinkedShapeAtom10 atoms follow to overwrite the heap so that [eax+8] points to an arbitrary location. As one can see, data issued from the file can be found in the stack around offset 0x00132B50. The program then jumps there to finally execute the payload.

This exploit however requires the victim to open the file by double clicking.

Detection

A LinkedSlideAtom10 may belong to four different containers named Handout, MainMaster, Notes or Slides. All are normally found in the Powerpoint Document stream.

In all cases, a file must be regarded malicious provided the size parameter of a LinkedSlideAtom10 is higher than 1FFFFFFFh which is for instance the case on Figure 1.

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
4:37F0h: 0C 00 04 00 00 00 00 00 00 00 02 00 0C 00 00 00  ................
4:3800h: E7 2E 08 00 00 00 01 00 00 00 01 00 00 20 00 00  ç.........… ..
4:3810h: E6 2E 08 00 00 00 01 00 00 00 41 41 41 41 00 00  æ.........AAAA..
4:3820h: E6 2E 08 00 00 00 01 00 00 00 41 41 41 41 00 00  æ.........AAAA..
4:3830h: E6 2E 08 00 00 00 01 00 00 00 41 41 41 41 00 00  æ.........AAAA..
4:3840h: E6 2E 08 00 00 00 01 00 00 00 41 41 41 41 00 00  æ.........AAAA..
```

Figure 1 – Malicious LinkedSlideAtom10

The malicious atom begins at offset 0x437FE. As one can see, 2EE7h stands for LinkedSlideAtom10. These atoms are normally 8 bytes long. Its first parameter is an index and the second represents the number of the LinkedShapeAtom atoms that follow. On Figure 1, this size equals 20000001h so this file is likely to trigger the vulnerability described here.

References

VUPEN/ADV-2009-1290:
http://www.vupen.com/english/advisories/2009/1290

CVE-2009-0221:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0221

Changelog

2009-05-20: Initial release