Abysssec Research

1) Advisory information

Title	: Microsoft Office Word sprmCMajority buffer overflow
Version	: Word 2007 SP 2
Analysis	: http://www.abysssec.com
Vendor	: http://www.microsoft.com
Impact	: Critical
Contact	: shahin [at] abysssec.com , info [at] abysssec.com
Twitter	: @abysssec
CVE	: CVE-2010-1900

2) Vulnerable version

Microsoft Works 9.0
Microsoft Word 2007 SP2
Microsoft Word 2007 SP1
Microsoft Word 2007 0
Microsoft Word 2003 SP3
Microsoft Word 2003 SP2
+ Microsoft Office 2003 SP1
+ Microsoft Office 2003 SP1
+ Microsoft Office 2003 0
+ Microsoft Office 2003 0
Microsoft Word 2003 SP1
+ Microsoft Office 2003 SP1
+ Microsoft Office 2003 SP1
+ Microsoft Office 2003 0
+ Microsoft Office 2003 0
Microsoft Word 2002 SP3
Microsoft Word 2002 SP2
+ Microsoft Office XP SP2

- Microsoft Windows 2000 Professional SP3
- Microsoft Windows 2000 Professional SP2
- Microsoft Windows 2000 Professional SP1
- Microsoft Windows 2000 Professional
- Microsoft Windows 98
- Microsoft Windows 98SE
- Microsoft Windows ME
- Microsoft Windows NT Workstation 4.0 SP6a
- Microsoft Windows NT Workstation 4.0 SP6
- Microsoft Windows NT Workstation 4.0 SP5
- Microsoft Windows NT Workstation 4.0 SP4
- Microsoft Windows NT Workstation 4.0 SP3
- Microsoft Windows NT Workstation 4.0 SP2
- Microsoft Windows NT Workstation 4.0 SP1
- Microsoft Windows NT Workstation 4.0
- Microsoft Windows XP Home SP1
- Microsoft Windows XP Home
- Microsoft Windows XP Professional SP1
- Microsoft Windows XP Professional

Microsoft Word 2002 SP1

- Microsoft Windows 2000 Professional SP2
- Microsoft Windows 2000 Professional SP1
- Microsoft Windows 2000 Professional
- Microsoft Windows 98
- Microsoft Windows 98SE
- Microsoft Windows ME
- Microsoft Windows NT Enterprise Server 4.0 SP6a
- Microsoft Windows NT Enterprise Server 4.0 SP6
- Microsoft Windows NT Enterprise Server 4.0 SP5
- Microsoft Windows NT Enterprise Server 4.0 SP4
- Microsoft Windows NT Enterprise Server 4.0 SP3
- Microsoft Windows NT Enterprise Server 4.0 SP2
- Microsoft Windows NT Enterprise Server 4.0 SP1
- Microsoft Windows NT Enterprise Server 4.0
- Microsoft Windows NT Server 4.0 SP6a
- Microsoft Windows NT Server 4.0 SP6
- Microsoft Windows NT Server 4.0 SP5
- Microsoft Windows NT Server 4.0 SP4
- Microsoft Windows NT Server 4.0 SP3
- Microsoft Windows NT Server 4.0 SP2
- Microsoft Windows NT Server 4.0 SP1
- Microsoft Windows NT Server 4.0
- Microsoft Windows NT Terminal Server 4.0 SP6
- Microsoft Windows NT Terminal Server 4.0 SP5
- Microsoft Windows NT Terminal Server 4.0 SP4
- Microsoft Windows NT Terminal Server 4.0 SP3
- Microsoft Windows NT Terminal Server 4.0 SP2
- Microsoft Windows NT Terminal Server 4.0 SP1

- Microsoft Windows NT Terminal Server 4.0 alpha
- Microsoft Windows NT Terminal Server 4.0
- Microsoft Windows NT Workstation 4.0 SP6a
- Microsoft Windows NT Workstation 4.0 SP6
- Microsoft Windows NT Workstation 4.0 SP5
- Microsoft Windows NT Workstation 4.0 SP4
- Microsoft Windows NT Workstation 4.0 SP3
- Microsoft Windows NT Workstation 4.0 SP2
- Microsoft Windows NT Workstation 4.0 SP1
- Microsoft Windows NT Workstation 4.0
- Microsoft Windows XP Home
- Microsoft Windows XP Professional
Microsoft Open XML File Format Converter for Mac 0
Microsoft Office Compatibility Pack 2007 SP2
Microsoft Office Compatibility Pack 2007 SP1
Microsoft Office Compatibility Pack 2007 0
Microsoft Office 2008 for Mac 0
Microsoft Office 2004 for Mac 0

3) Vulnerability information

Class

1- stack overflow

Impact

An attacker can exploit this issue to execute arbitrary code in the context of the currently logged-in user. Failed exploit attempts will likely result in denial-of-service conditions.

Remotely Exploitable

Yes

Locally Exploitable

Yes

4) Vulnerabilities detail

This vulnerability show itself in processing of sprmCMajority record. Because of not checking parameters when processing sprm groups related to sprmCMajority, it is possible to control amount of buffer that should be copied to the buffer, and as a result an stack overflow.

wwlib.dll module is responsible for processing sprm group. And function sub_31c1f0f2 of this module is responsible for processing sprmCMajority record.

In the beginning of this function, value of FIB.FibBase.nFib field is checked if less than 0x4E or not. In case value of this field greater than 0x4E, sprmCMajority record parameter is copied to a buffer. Length of the parameter is specified by the first byte after sprmCMajority record code (0xCA47). Maximom length of this record is 255 bytes:

```
.text:31C1F0F2          push    ebp
.text:31C1F0F3          mov     ebp, esp
.text:31C1F0F5          sub     esp, 78Ch
.text:31C1F0FB          mov     eax, ds:dword_31469FC0
.text:31C1F100          xor     eax, ebp
.text:31C1F102          mov     [ebp+var_4], eax
.text:31C1F105          cmp     [ebp+arg_8], 4Eh
.text:31C1F109          push    ebx
.text:31C1F10A          push    esi
.text:31C1F10B          mov     esi, [ebp+arg_0]
.text:31C1F10E          mov     eax, [esi]
.text:31C1F110          movzx   ebx, byte ptr [eax]
.text:31C1F113          push    edi
.text:31C1F114          mov     edi, [ebp+arg_4]
.text:31C1F117          push    ebx             ; Size
.text:31C1F118          lea     ecx, [eax+1]    ; Src
.text:31C1F11B          jl      short loc_31C1F16E
.text:31C1F11D          lea     edx, [ebp+Src]  ; Dst
.text:31C1F123          call    sub_312498A0
```

sub_312498A0 function does the copying task. Then sub_31C1B83E is called, this function is responsible for processing sprm groups of sprmCMajority record.

```
.text:31C1F128          push    0FFFh
.text:31C1F12D          push    3BBh
.text:31C1F132          push    [ebp+arg_18]
.text:31C1F135          movzx   eax, bx
```

```
.text:31C1F138          push    [ebp+arg_14]
.text:31C1F13B          mov     [ebp+var_780], eax
.text:31C1F141          push    [ebp+arg_10]
.text:31C1F144          lea     eax, [ebp+Src]
.text:31C1F14A          push    0
.text:31C1F14C          push    [ebp+arg_C]
.text:31C1F14F          push    [ebp+arg_8]
.text:31C1F152          push    eax
.text:31C1F153          lea     eax, [ebp+var_780]
.text:31C1F159          push    eax
.text:31C1F15A          mov     eax, [esi]
.text:31C1F15C          inc     eax
.text:31C1F15D          push    eax
.text:31C1F15E          call    sub_31C1B83E
```

In the doc file format documents the following elements are mentioned as attributes that sprmCMajority can have effects on them:

```
chp.fBold,
chp.fItalic,
chp.fSmallCaps,
chp.fVanish,
chp.fStrike,
chp.fCaps,
chp.rgftc,
chp.hps,
chp.hpsPos,
chp.kul,
chp.dxaSpace,
chp.ico,
chp.rglid
chp.fOutline
chp.fShadow
chp.ftc
chp.cv
```

But actually sub_31C1B83E function, also processes other sprm groups which affect other properties. One of these sprm groups, is sprmPAnld80 (0xC63E). the following code from function sub_31C1B83E is responsible for processing this sprm.
In the beginning of this code two comparison is performed. First comparison is related to seventh argument of this function. If value of the argument is not equal to zero, the second comparison is will be performed. In the second comparison value of FIB.FibBase.nFib field is compared with A4 constant and if less the sub_31284D06 function will be called. This function examines value of IB.FibBase.wIdent field. Value of this field in word files is equal to 0xA5EC. If value of this field is valid the function returns zero.

```
.text:31C1CB2B          cmp     [ebp+arg_18], 0
.text:31C1CB2F          jz      loc_31C1D7F0
.text:31C1CB35          cmp     [ebp+arg_C], 0A4h
.text:31C1CB3C          jge     loc_31C1D7F0
.text:31C1CB42          movzx   eax, word ptr [ebp+arg_1C]
.text:31C1CB46          push    eax
.text:31C1CB47          call    sub_31284D06
.text:31C1CB4C          test    eax, eax
.text:31C1CB4E          jnz     loc_31C1D7F0
```

If our doc file passes these conditions, it reaches our vulnerable code. In this part of the code, first parameter of sprmPAnld80 is copied to edi register. Then for 84bytes of buffer in stack is initialized to zero with memset call. Then for the amount of sprmPAnld80 parameter, our data is copied to this 84bytes byffer by calling sub_312498A0 function.
The vulnerable point of this code is lack of checking sprmPAnld80 paramter. In case of greater than 84 for this parameter, buffer overflow occurs.

```
.text:31C1CB5F          mov     eax, [ebp+var_160]
.text:31C1CB65          movzx   edi, byte ptr [eax]
.text:31C1CB68          inc     [ebp+var_160]
.text:31C1CB6E          push    54h             ; Size
.text:31C1CB70          lea     eax, [ebp+Dst]
.text:31C1CB73          push    0               ; Val
.text:31C1CB75          push    eax             ; Dst
.text:31C1CB76          call    memset
.text:31C1CB7B          mov     eax, [ebp+var_15C]
.text:31C1CB81          add     esp, 0Ch
.text:31C1CB84          mov     byte ptr [eax], 54h
.text:31C1CB87          mov     ecx, [ebp+var_160] ; Src
.text:31C1CB8D          inc     [ebp+var_15C]
.text:31C1CB93          push    edi             ; Size
.text:31C1CB94          lea     edx, [ebp+Dst]  ; Dst
.text:31C1CB97          call    sub_312498A0
```

Exploit

Vulnerability is a stack based overflow and we can overwrite return address. But because of GS protection, by overwriting Return address, the program will be terminated. If you able to overwrite SEH structure and cause and exception then it is possible to bypass this protection and take the execution flow.