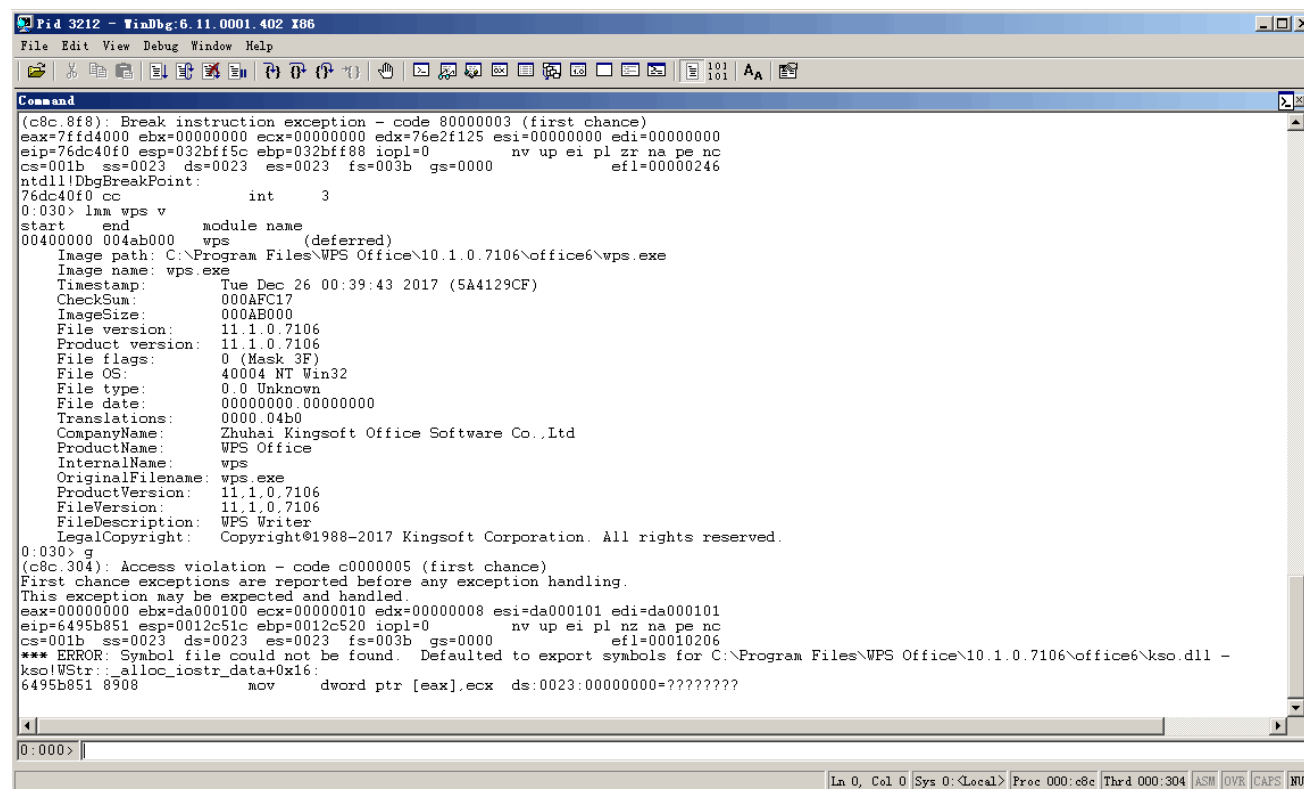


An issue was discovered in WPS Office

Here is an issue in WPS Office 10.2.0.5978 and 10.1.0.7106, and possibly have impacted other versions.

Remote attackers could leverage this vulnerability to cause a denial of service via a crafted (a) web page, (b) office document, or (c) .rtf file.

It was discovered in the module of kso.dll.



```
Pid 3212 - WinDbg: 6.11.0001.402 x86
File Edit View Debug Window Help
[Icons]
Command
(c8c.8f8): Break instruction exception - code 80000003 (first chance)
eax=76fd4000 ebx=00000000 ecx=00000000 edx=76e2f125 esi=00000000 edi=00000000
eip=76dc40f0 esp=032bff5c ebp=032bff88 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000246
ntdll!DbgBreakPoint:
76dc40f0 cc                int     3
0:030> lnm wps v
start  end      module name
00400000 004ab000 wps (deferred)
Image path: C:\Program Files\WPS Office\10.1.0.7106\office6\wps.exe
Image name: wps.exe
Timestamp: Tue Dec 26 00:39:43 2017 (5A4129CF)
CheckSum: 000AFC17
ImageSize: 000AB000
File version: 11.1.0.7106
Product version: 11.1.0.7106
File flags: 0 (Mask 3F)
File OS: 40004 NT Win32
File type: 0.0 Unknown
File date: 00000000.00000000
Translations: 0000.04b0
CompanyName: Zhuhai Kingsoft Office Software Co.,Ltd
ProductName: WPS Office
InternalName: wps
OriginalFilename: wps.exe
ProductVersion: 11.1.0.7106
FileVersion: 11.1.0.7106
FileDescription: WPS Writer
LegalCopyright: Copyright©1988-2017 Kingsoft Corporation. All rights reserved.
0:030> g
(c8c.304): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=da000100 ecx=00000010 edx=00000008 esi=da000101 edi=da000101
eip=6495b851 esp=0012c51c ebp=0012c520 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010206
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\Program Files\WPS Office\10.1.0.7106\office6\kso.dll -
kso!WStr::_alloc_iostr_data+0x16:
6495b851 8908                mov     dword ptr [eax],ecx  ds:0023:00000000=????????
0:000>
```

At the crash point, calculating its offset from the starting address of kso.dll.

```
0:000> ? 6495b851 - kso
Evaluate expression: 309329 = 0004b851
```

Using IDA to examine the codes. When it called the malloc function, without any check, the memory address of the return value was been written. Causing access violation.

```

.text:1004B83B
.text:1004B83B ; ===== S U B R O U T I N E =====
.text:1004B83B
.text:1004B83B ; Attributes: bp-based frame
.text:1004B83B
.text:1004B83B ; struct WStr::iostr_data *__cdecl WStr::_alloc_iostr_data(unsigned int)
.text:1004B83B         public ?_alloc_iostr_data@WStr@@CAPAUiostr_data@1@I@Z
.text:1004B83B ?_alloc_iostr_data@WStr@@CAPAUiostr_data@1@I@Z proc near
.text:1004B83B                                     ; CODE XREF: WStr::WStr(ushort const *)+261p
.text:1004B83B                                     ; WStr::assign(ushort const *)+3B1p ...
.text:1004B83B
.text:1004B83B arg_0             = dword ptr  8
.text:1004B83B
.text:1004B83B         push     ebp
.text:1004B83C         mov      ebp, esp
.text:1004B83E         push     esi
.text:1004B83F         mov      esi, [ebp+arg_0]
.text:1004B842         lea      eax, [esi+esi+10h]
.text:1004B846         push     eax             ; size_t
.text:1004B847         call    ds:__imp_malloc
.text:1004B84D         pop      ecx
.text:1004B84E         lea      ecx, [eax+10h]
.text:1004B851         mov      [eax], ecx
.text:1004B853         lea      ecx, [ecx+esi*2]
.text:1004B856         mov      [eax+8], esi
.text:1004B859         mov      [eax+4], ecx
.text:1004B85C         mov     dword ptr [eax+0Ch], 1
.text:1004B863         pop      esi
.text:1004B864         pop      ebp
.text:1004B865         retn
.text:1004B865 ?_alloc_iostr_data@WStr@@CAPAUiostr_data@1@I@Z endp
.text:1004B865

```