

Microsoft Office Protected-View Out-Of-Bound Array Access

2017-11-23

Software	Microsoft Office
Affected Versions	Microsoft Excel 2010, 2013, 2016 (x86 and x64)
CVE Reference	CVE-2017-8692 (Uniscribe Remote Code Execution Vulnerability)
Author	Yong Chuan Koh (@yongchuank)
Severity	Important
Vendor	Microsoft Corporation
Vendor Response	Fixed on 12 Sept 2017

Description:

Microsoft Office is a suite of desktop applications consisting of Word, Excel, Powerpoint, Outlook and various other productivity applications. Among these, Word, Excel and Powerpoint implemented the Protected-View sandbox technology as a defence-in-depth exploit mitigation.

An out-of-bound array access was discovered while the Excel broker parsed an attacker controlled Protected-View Inter-Process Communication (IPC) message from the sandbox process.

Impact:

A successful exploitation would allow an attacker to elevate his privileges from AppContainer to Medium, thereby breaking out of the Protected-View sandbox.



Cause:

The vulnerability existed because as the broker process looped through an array of SCRIPT_ITEM objects, it dereferences the current (N) and next (N+1) SCRIPT_ITEM objects to calculate the difference of iCharPos value between these two objects. However, if N is the last SCRIPT_ITEM object, then an out-of-bound dereference for the N+1 object would occur.

Interim Workaround:

Avoid opening Microsoft Office Excel files from untrusted sources.

Solution:

Users should apply the September security updates from Microsoft.

Technical details

The following analysis is based on EXCEL 16.0.4266.1001.

In Protected-View mode, the EXCEL broker receives and services IPC messages from the Excel sandbox, distinguished by a message-tag. One of these messages has the 0x071200 tag, which the sandbox uses to request the broker to input a specified string into the formula bar as it is isolated from the AppContainer. Subsequently, the Excel broker would process the formula-bar string with gdi32full!ScriptItemize().

This POC formula-bar string is sent to Excel broker with the 0x071200 IPC message:

```
?0S^'&1:bCWX4[tcY%=D~W@vJ}MpMr<ijSar<#9<OtrX_S7j\ldH"?qF>!uMnO>(q-j(@-
g?Mcav)MzM_<m+T[zA46ykl#V5\2Kj|42</pre>
```

Upon receipt, Excel would transform the string into an array of SCRIPT_ITEM objects with the following sequence of calls to gdi32full!ScriptItemize().

```
ScriptItemize (
```

- pwcInChars =
 L"?0S^'&1:bCWX4[tcY%=D~W@vJ}MpMr<ijSar<#9<OtrX_S7j\ldH"?qF>!uMnO>(q-j(@g?Mcav)MzM <m+T[zA46ykI#V5\2Kj|42"</pre>
- cInChars = 0x0000021
- cMaxItems = 0x00000022



```
- psControl = &(0x00800009)
   - psState = & (0x0001)
   - pItems
   - pcItems
)
ScriptItemize (
   - pwcInChars = L"Sar<#9<OtrX S7j\ldH"?qF>!uMnO>(q-j(@-
     g?Mcav)MzM <m+T[zA46ykI#V5\2Kj|42"
   - cInChars = 0x0000021
   - cMaxItems = 0x0000022
   - psControl = \& (0x00800009)
   - psState = \& (0x0001)
   - pItems
   - pcItems
)
ScriptItemize (
   - pwcInChars = L"j(@-g?Mcav)MzM <m+T[zA46ykI#V5\2Kj|42"</pre>
   - cInChars = 0x0000004
   - cMaxItems = 0x00000005
   - psControl = & (0x00800009)
   - psState = & (0x0001)
   - pItems
   - pcItems
)
```

As the out-of-bound dereference occurred in the last call to gdi32full!ScriptItemize() on the *pItems* output buffer, we next examine how it was allocated. After some reversing, the buffer was found to be allocated in Mso99Lwin32client!sub_B6899(), together with an assignment of the *cMaxItems* parameter. The snippet below shows the relevant blocks of this allocation.



		* * * * *		
5A25690A	1			
5A25690A 5A25690A	lea	6690A: ; eax, [ebp+cMaxItems] ; at this point, esi = cMaxItems = cInChars = 4		
5A25690D 5A25690E	push	eax esi ; cInChars		
5A25690F	call	CopyArguments		
5A256914 5A256918		[ebp+var_38], 0 eax, [ebp+var_38]		
5A25691B 5A25691C	push	eax		
5A25691F	MOV	eax, [ebp+var_3C] [ebp+var_3C], 4		
5A256926 5A256927		eax eax [ebp+cMaxItems]		
5A25692A 5A25692B	push	eax		
5A256930	mov	MultiplyArguments eax, [ebp+var_38] ; var_38 = cMaxItems * 4 = 4 * 4 = 10h [ebp+var_38], 0		
5A256933 5A256937		[ebp+var_38], 0 [ebp+cMaxItems], eax ; cMaxItems = 10h		
5A25693A 5A25693D		eax, [ebp+var_38] eax		
5A25693E	lea	eax, [ebp+var_3C]		
5A256941 5A256948		[ebp+var_3C], 3 eax		
5A256949 5A25694C	lea	eax, [ebp+cMaxItems] eax		
5A25694D	call	MultiplyArg0Arg4		
5A256952 5A256955	push	eax, [ebp+var_38] ; var_38 = cMaxItems * 03h = 10h * 03h = 30h 2		
5A256957 5A256958	pop	ecx [ebp+var_3C], eax ; var_3C = var_38 = 30h		
5A25695B	lea	eax, [esi+1] ; eax = cInChars + 1 = 4 + 1 = 5		
5A25695E 5A256960		eax, ecx 8		
5A256962 5A256962	cmovg	ecx, eax ; ecx = eax > ecx ? eax : ecx ; = (cInChars+1) > 2 ? (cInChars+1) : 2		
5A256962		; = (cInChars+1) = 5		
5A256962 5A256965	lea	eax, [ebp+var 40]		
5A256968 5A256969	push	ecx eax		
5A25696A	MOV	[ebp+cMaxItems], ecx		
5A25696D 5A256972		MultiplyArg0Arg4_Wrapper ecx, [ebp+var_3C] ; [ecx] = var_3C = 30h		
5A256975 5A256977	push	dword ptr [eax] ; [eax] = cMaxItems * 08h = 05h * 08h = 28h eax, [ebp+dwBufferSize]		
5A25697A	push	eax		
5A25697B 5A256980		AddArguments_Wrapper [ebp+ppBuffer]		
5A256983 5A256985		<pre>[ebp+ppborrer] dword ptr [eax] ; [eax] = dwHeapAllocSize = 28h + 30h = 58h HeapAlloc_Wrapper</pre>		
5A25698A 5A25698C	test	<pre>eax, eax ;> this buffer = [unknown_buffer]+[pcItems_buffer] loc_5A256AD1</pre>		
		5A256992 lea eax, [ebp+dwBufferSize] 5A256995 push eax		
		5A256996 push esi		
		5A256997 call CopyArguments 5A25699C and [ebp+var_38], 0		
		5A2569A0 push 20h		
		5A2569A2 pop eax 5A2569A3 mov [ebp+var_3C], eax		
		5A2569A6 lea eax, [ebp+var_38] 5A2569A9 push eax		
		5A2569AA lea eax, [ebp+var_3C] 5A2569AD push eax		
		5A2569AE lea eax, [ebp+dwBufferSize]		
		5A2569B1 push eax 5A2569B2 call MultiplyArguments		
		5A2569B7 push [ebp+arg_34] 5A2569BA push [ebp+var_38]		
		5A2569BD call HeapAlloc_Wrapper		
		5A2569C2 mov [ebp+var_3C], eax 5A2569C5 test eax, eax		
		5A2569C7 mov eax, [ebp+ppBuffer] 5A2569CA js loc_5A265553		
		5A2569D0 mov ecx, [eax] ; ecx = pBuffer		
		5A2569D2 lea eax, [ecx+esi*8] ; esi = cInChars = 4 5A2569D5 mov [ebp+var_C], eax ; pBuffer+(cInChars*8)		
		5A2569D8 lea edx, [ecx+esi*4]		
		E02E60DD imul oby oci 80b $+$ oby $-$ clathorewsky		
		5A2569DB imul eax, esi, 0Ch ; eax = cInChars*0xC 5A2569DE mov [ebp+var_14], edx ; pBuffer+(cInChars*4)		
		5A2569DB imul eax, esi, 0Ch ; eax = clnChars*0xC 5A2569DE mov [ebp+var_14], edx ; pBuffer+(clnChars*4) 5A2569E1 add eax, ecx ; eax = pBuffer+(clnChars+0xC)		
		5A2569DB imuleax, esi, 0Ch; eax = clnChars*0xC5A2569DE mov[ebp+var_14], edx ; pBuffer+(clnChars*4)5A2569E1 addeax, ecx; eax = pBuffer+(clnChars*0xC)5A2569E1: = pItems5A2569E3 movecx, [ebp+ppItems]		
		5A2569DB imuleax, esi, 0Ch ; eax = clnChars*0xC5A2569DE mov[ebp+var_14], edx ; pBuffer+(clnChars*4)5A2569E1 addeax, ecx ; eax = pBuffer+(clnChars*0xC)5A2569E1; = pItems5A2569E3 movecx, [ebp+ppItems]5A2569E6 mov[ecx], eax5A2569E8 moveax, [ebp+arg_50]		
		<pre>5A2569DB imul eax, esi, 0Ch ; eax = clnChars*0xC 5A2569DE mov [ebp+var_14], edx ; pBuffer+(clnChars*4) 5A2569E1 add eax, ecx ; eax = pBuffer+(clnChars*4) 5A2569E1 ; = pItems 5A2569E3 mov ecx, [ebp+ppItems] 5A2569E6 mov [ecx], eax 5A2569E8 mov eax, [ebp+arg_50] 5A2569E8 test eax.</pre>		
		5A2569DB imuleax, esi, 0Ch ; eax = clnChars*0xC5A2569DE mov[ebp+var_14], edx ; pBuffer+(clnChars*4)5A2569E1 addeax, ecx ; eax = pBuffer+(clnChars*0xC)5A2569E3 movecx, [ebp+ppItems]5A2569E6 mov[ecx], eax5A2569E8 moveax, [ebp+arg_50]		

Figure 1: Snippet in Mso99Lwin32client!sub_B6899(), to allocate pltems buffer



In the above snippet, at Mso99Lwin32client!000B6962 with a *clnChars* value of 4, the *cMaxItems* parameter is determined by the following:

• *cMaxChars* = *clnChars* > 2 ? (*clnChars*+1) : 2

Next, in the code-block Excel allocates a buffer (*pBuffer*) of 58h bytes consisting of these 2 sub-buffers:

- Size of Unknown-SubBuffer = *clnChars* * 4 * 3
 = 30h bytes
- Size of pltems-SubBuffer = (*cMaxItems*) * sizeof(SCRIPT_ITEM)
 - = (*cMaxItems*) * 8 = 28h bytes

Eventually a pointer to *pltems* is obtained from *pBuffer*.

pltems = pBuffer + (clnChars * 0xC)
 = pBuffer + (clnChars * 3 * 4)

Finally, back in gdi32full!ScriptItemize(), it was observed that it called gdi32full!ScriptTokenize() to write *clnChars* number of SCRIPT_ITEM structures retrieved from the *pwclnChars* parameter to the *pltems* buffer. The last SCRIPT_ITEM (ie: *cMaxItems*-th) object is then used to "summarize" the number of tokenized characters with the flag 3.

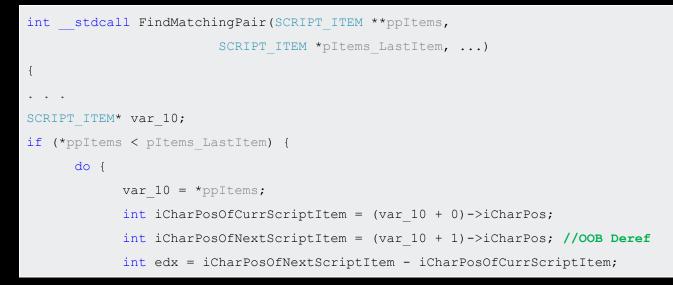
The snippet below shows the code-blocks where the last SCRIPT_ITEM object is written:



7430BDD1 cmp 7430BDD1 cmp 7430BDD1 cmp 7430BDD4 j1			
▲ 2430800A lea 7430800F push 7430800F push 743080E9 push 743080E5 push 743080E5 push 743080E6 call 743080EF cap 743080EF cnp 743080EF cnp 743080EF cnp	edx, [esp+30h+var_18]; OUT_ARG of ScriptTokenize(), which is clnChars edx ; int edi ; pltens [ebp+const_NULL]; int edx, eax esi ; int cHaxItens ?ScriptTokenize@@PGJPB_UHHHPAUTOKEN@@PAH#22; ScriptTokenize(wchar_t const *,int,int,int,TOKEN *,int *) ecx, [esp+30h+var_18]; OUT_ARG of ScriptTokenize(), which is clnChars ecx, esi loc_7430060B		
		74308DF7 mov 74308DF7 mov 74308DFA lea 74308DFA 74308DF0 mov 74308E01 mov 74308E01 mov 74308E03 mov 74308E05 cest 74308E06 jz	ebx, [ebp+psControl] esi, [edi+ecx*8] ; edi = pItens ; esi = (SCRIPT_ITEN*)(pItens+cInChars) ecx, [esp+30h+cInChars] [esi], ecx ecx, 3 [esi+6], cx ebx, ebx short loc_74300E61

Figure 2: Snippet in gdi32full!ScriptItemizeCommon(), to populate pltems buffer

The populated *pltems* buffer is then parsed into a loop in gdi32full!FindMatchingPair(), which is represented by the following pseudo-code:





```
if (var_10->a.eScript > 0x114) { ... }
else if (var_10->a.eScript == 0x114) { ... }
else { ... }
(*ppItems)++;
} while (*ppItems <= pItems_LastItem);
}
...
}</pre>
```

In this do-while loop, the *pltems* pointer is incremented until the last SCRIPT_ITEM object. However in the loop, the next SCRIPT_ITEM object is also dereferenced. Therefore this causes an out-of-bound dereference when *pltem* points to the last SCRIPT_ITEM object.

The following windbg output show below demonstrates the crash:



0:000) g (249c.1958): Access violation - code c0000005 (!!! second chance !!!) eax=46412ff8 ebx=00000004 ecx=46412ff8 edx=00000000 esi=00000000 eip=7431ees3 esp=00366420 ebp=0036643c iop1=0	\Office1(
<pre>eip-7431ee9e esp-00306d20 ebp=00306d3c iopl=0</pre>	\Office10
<pre>gdi32full+FindMatchingPair+0x30:</pre>	\Office10
0.000> theap -p -a eax address 46442f8f found in DFH_HEAP_ROOT @ 3001000 in busy allocation (DFH_HEAP_ELOCK: UserAddr UserSize - VirtAddr VirtSize) 46521b94: 464f2fa8 58 - 464f2000 2000 6d459c2c verifier AVrfDeugPageHeapAllocate+0x0000023c 7776120 ntdll RtlDebugAllocateHeap+0x000001642 776f1431 ntdll RtlpAllocateHeap+0x00000124 776f1931 ntdll RtlpAllocateHeap+0x0000002a *** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Common Files\Microsoft Shared 5b172457 msc20vin32client1Ordinal1151+0x00000079 5b172457 msc20vin32client1Ordinal1151+0x00000079 5b1723ec msc20vin32client1Ordinal1151+0x00000079 5b1723a0 msc20vin32client+0rdinal154-0x00000079 5b1723a0 msc20vin32client+0x000b6b09 5a256909 msc991vin32client+0x0013b58a 0085ea5f EXCEL1Ordinal43+0x0015056 0030de55 EXCEL1Ordinal43+0x0015056 00430b5f EXCEL1Ordinal43+0x00050b5f 00430b5f EXCEL1Ordinal43+0x00050b5f 00466fb9 EXCEL1Ordinal43+0x00050b5f 00430b5f EXCEL1Ordinal43+0x00050b5f 00430b5f EXCEL1Ordinal43+0x00050b5f 00446fb9 EXCEL1Ordinal43+0x00050b5f 00446fb9 EXCEL1Ordinal43+0x00050b5f 00446fb9 EXCEL1Ordinal43+0x00050b5f 00446fb9 EXCEL1Ordinal43+0x00050b5f 0053e45c EXCEL1Ordinal43+0x00050b5f 0053e45c EXCEL1Ordinal43+0x00050b5f 0053e45c EXCEL1Ordinal43+0x00050b5f 0053e45c EXCEL1Ordinal43+0x00050b5f 0053e45c EXCEL1Ordinal43+0x00050b5f 0053e45c	\Office10
DPH_HEAP_ROOT @ 3001000 in busy allocation (DFH_HEAP_BLOCK: UserAddr UserSize - VirtAddr VirtSize) 46521b94: 464f21b38 58 - 464f2000 2000 6d455c2c verifier!AVrtBobugBageHeapAllocate40x000003c 777a0d2e ntdl!RtlpAllocateHeap+0x00000042 776f193 ntdl!RtlpAllocateHeap+0x00000024 776f193 ntdl!RtlpAllocateHeap+0x00000024 *** EERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Common Files\Microsoft Shared 5b172457 msc20vin32client!Ordinal1151+0x00000027 5b172457 msc20vin32client!Ordinal1151+0x00000079 5b1723ec msc20vin32client!Ordinal1151+0x00000079 5b1723ec msc20vin32client!Ordinal151+0x00000026 5a256b9 msc99Iwin32client+0x000b609a Sa2da51d msc99Iwin32client+0x000b608a Sa2da51d msc99Iwin32client+0x0013a51d 5a2db58m msc99Iwin32client+0x0013e58 0085deff EXCEL!Ordinal43+0x00050b5f 00430b5f EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 003e194 EXCEL!Ordinal43+0x00050b5f 005e30 EXCEL!Ordinal43+0x00050b5f	\Office10
<pre>in busy allocation (DPH_HEAP_BLOCK: UserAddr UserSize - VirtAddr VirtSize)</pre>	\Office10
<pre>6d459c2c verifier/AVrtDebugPageHeapAllocate+0x000003c 776f4732 ntdllRtlpAllocateHeap+0x00000042f 776f19ds ntdllRtlpAllocateHeap+0x000002a **** EEROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Common Files\Microsoft Shared 5b172457 msc20win32client!Ordinal1151+0x0000002 5b172459 msc20win32client!Ordinal1151+0x0000003c 5b172429 msc20win32client!Ordinal1151+0x0000003c 5b1723ec msc20win32client!Ordinal1151+0x0000003c 5b1723ec msc20win32client!Ordinal1151+0x00000026 5a25691 msc91win32client+0x0006b09 5a25693 msc91win32client+0x000b6b09 5a26593 msc91win32client+0x0013a51d 5a2d558 msc99Iwin32client+0x0013b58a 0085ea9 EXCEL!Ordinal43+0x0047ea9f 0050de55 EXCEL!Ordinal43+0x0047ea9f 00430b5f EXCEL!Ordinal43+0x00050b5f 00430b5f EXCEL!Ordinal43+0x00050b5f 00430b5f EXCEL!Ordinal43+0x00050b5f 00430b5f EXCEL!Ordinal43+0x0008fb9 00466fb9 EXCEL!Ordinal43+0x0008fb9 00466fb9 EXCEL!Ordinal43+0x0008fb9 0046883 EXCEL!Ordinal43+0x0008fb9 0046883 EXCEL!Ordinal43+0x0008fb9 0046883 EXCEL!Ordinal43+0x0009dec 003a1194 EXCEL!Ordinal43+0x0009dec 003a1194 EXCEL!Ordinal43+0x0009dec 003a1194 EXCEL!Ordinal43+0x0000002b 77719b3 ntdl!!_RtlUserThreadStart+0x000001b</pre>	\Office10
<pre>777a0d2e ntdl!RtlDebugAllocateHeap+0x000003c 776f193 ntdl!RtlpAllocateHeapInternal+0x0000042f 776f193 ntdl!RtlpAllocateHeapInternal+0x000002a *** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Common Files\Microsoft Shared 5b172457 msc20win32client!Ordinal1151+0x0000002 5b1723e msc20win32client!Ordinal1151+0x0000003 5b1723e msc20win32client!Ordinal1151+0x00000026 5a255b09 msc99Lwin32client!Ordinal1675+0x0000026 5a255b18 msc99Lwin32client+0x000b659a 5a26548 msc99Lwin32client+0x000b659a 5a26548 msc99Lwin32client+0x0013a51d 5a2d518 msc99Lwin32client+0x0013b58a 0085ea9f EXCEL!Ordinal43+0x0047ea9f 0050de55 EXCEL!Ordinal43+0x0050b5f 00430b5f EXCEL!Ordinal43+0x0050b5f 00430b5f EXCEL!Ordinal43+0x0050b5f 00430b5f EXCEL!Ordinal43+0x0050b5f 00430b5f EXCEL!Ordinal43+0x00050b5f 00430b5f EXCEL!Ordinal43+0x0008fb9 0040883 EXCEL!Ordinal43+0x0008fb9 00466fb9 EXCEL!Ordinal43+0x0008fb9 0046883 EXCEL!Ordinal43+0x0008fb9 004888 EXCEL!Ordinal43+0x0008fb9 004888 EXCEL!Ordinal43+0x0001485c 003e445c EXCEL!Ordinal43+0x0000194 75388e94 KERNEL32!BaseThreadInitThunk+0x00000024 77719b3 ntdl!!_RtlUserThreadStart+0x000001b</pre>	\Office1\
<pre>776fle3f ntdll!RtlplilocateHeapInternal+0x000002a 776fle3f ntdll!RtlplilocateHeapInternal+0x000002a *** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Common Files\Microsoft Shared 5b172457 mso20win32client!Ordinal1151+0x0000003 5b1723ec mso20win32client!Ordinal1151+0x0000003c 5b1723e mso20win32client!Ordinal1151+0x00000026 5a25598 mso99Iwin32client+0x000b6090 5a25598 mso99Iwin32client+0x000b6090 5a25598 mso99Iwin32client+0x0013a51d 5a2d518 mso99Iwin32client+0x0013b58a 0085a9f EXCEL!Ordinal43+0x0012d55 00430b9f EXCEL!Ordinal43+0x00050b97 00430b5f EXCEL!Ordinal43+0x00050b97 00430b5f EXCEL!Ordinal43+0x00050b97 00430b5f EXCEL!Ordinal43+0x00050b97 0046fb9 EXCEL!Ordinal43+0x00050b97 0046883 EXCEL!Ordinal43+0x00050b97 0046883 EXCEL!Ordinal43+0x00050b97 00430b5f EXCEL!Ordinal43+0x00050b97 00430b5f EXCEL!Ordinal43+0x00050b97 00466fb9 EXCEL!Ordinal43+0x00050b97 00466fb9 EXCEL!Ordinal43+0x00050b97 00466fb9 EXCEL!Ordinal43+0x00050b97 003e4a5c EXCEL!Ordinal43+0x00050b92 003e4a5c EXCEL!Ordinal43+0x00050b92 003e4a5c EXCEL!Ordinal43+0x00050b92 003e4a5c EXCEL!Ordinal43+0x00004a5c 003e145 KERNEI32!BaseThreadInitThunk+0x00000024 77719b3 ntdl!!_RtlUserThreadStart+0x000001b</pre>	∖Office1
<pre>*** ERROR: Symbol file could not be found. Defaulted to export symbols for C:\Program Files\Common Files\Microsoft Shared</pre>	∖Office1
<pre>5b172457 mso20win32client!Ordinal1151+0x0000007 5b1723e mso20win32client!Ordinal1151+0x00000079 5b1723e mso20win32client!Ordinal1151+0x0000026 5a25698 mso99Iwin32client+0x0006698 5a26598 mso99Iwin32client+0x001698a 5a2da51d mso99Iwin32client+0x0013a51d 5a2db58 mso99Iwin32client+0x0013b58a 0085e3F EXCEL!Ordinal43+0x0047ea9f 0050de55 EXCEL!Ordinal43+0x0047ea9f 00430b5f EXCEL!Ordinal43+0x0050b5f 00430b5f EXCEL!Ordinal43+0x00050b5f 00430b5f EXCEL!Ordinal43+0x00050b5f 00430b5f EXCEL!Ordinal43+0x000509dc 00466fb9 EXCEL!Ordinal43+0x00050b5f 00466fb9 EXCEL!Ordinal43+0x00050b5f 00466fb9 EXCEL!Ordinal43+0x00050b5f 003e4a5c EXCEL!Ordinal43+0x00050b5f 003e4a5c EXCEL!Ordinal43+0x0000485c 003e1194 EXCEL!Ordinal43+0x0000485c 003e1194 EXCEL!Ordinal43+0x0000485c 003e1194 EXCEL!Ordinal43+0x00000024 77719bc3 ntdl!!_RtlUserThreadStart+0x000001b</pre>	
<pre>5h723ec mso20win32client10rdinal1151+0x0000003c 5h1723ec mso20win32client10rdinal675+0x0000026 5a256b09 mso99Iwin32client+0x000b680a 5a2ds50a mso99Iwin32client+0x0013a51d 5a2ds50a mso99Iwin32client+0x0013a51d 5a2ds50a mso99Iwin32client+0x0013b50a 0085ea9f EXCEL10rdinal43+0x0017ea9f 0050de55 EXCEL10rdinal43+0x0017ea9f 00430b5f EXCEL10rdinal43+0x00050b5f 00430b5f EXCEL10rdinal43+0x00050b5f 00430b5f EXCEL10rdinal43+0x000509dc 00466fb9 EXCEL10rdinal43+0x000509dc 00466fb9 EXCEL10rdinal43+0x00086fb9 0040883 EXCEL10rdinal43+0x000485c 003ed35c EXCEL10rdinal43+0x000485c 003ed35c EXCEL10rdinal43+0x000485c 003ed35c EXCEL10rdinal43+0x000194 76388e94 KERNEL321BaseThread5tart+0x00000024 77719bc3 ntdl11Rt1UserThreadStart+0x000001b</pre>	
<pre>5b1723a0 mso20win32client!Ordinal675+0x0000026 5a25698 mso99Iwin32client+0x000b699 5a26598 mso99Iwin32client+0x0013a51d 5a2db58 mso99Iwin32client+0x0013b58a 0085ea9f EXCEL!Ordinal43+0x0012de55 00430b57 EXCEL!Ordinal43+0x00050b97 00430b5f EXCEL!Ordinal43+0x00050b5f 004309de EXCEL!Ordinal43+0x00050bf 004309de EXCEL!Ordinal43+0x00050bf 00466fb9 EXCEL!Ordinal43+0x00050bf 00466fb9 EXCEL!Ordinal43+0x00050bf 0046838 EXCEL!Ordinal43+0x00050bf 00466fb9 EXCEL!Ordinal43+0x00050bf 00466fb9 EXCEL!Ordinal43+0x0002883 003e4a5c EXCEL!Ordinal43+0x0004a5c 003e1454 EXCEL!Ordinal43+0x00004a5c 003e1454 EXCEL!Ordinal43+0x0001194 76388e94 KERNEL32!BaseThreadInitThunk+0x0000024 77719bc3 ntdl!RtlUserThreadStart+0x000002b 77719b92 ntdl!_RtlUserThreadStart+0x000001b</pre>	
5a26598a msc99Lwin32client+0x000b698a 5a2ds51d msc99Lwin32client+0x0013b58a 0085ea9f EXCEL!Ordinal43+0x0012de55 00430b97 EXCEL!Ordinal43+0x0012de55 00430b5f EXCEL!Ordinal43+0x00050b5f 004309dc EXCEL!Ordinal43+0x000509dc 00466fb9 EXCEL!Ordinal43+0x000509dc 00466fb9 EXCEL!Ordinal43+0x0008fb9 00408883 EXCEL!Ordinal43+0x0002883 003e4a5c EXCEL!Ordinal43+0x0002883 003e4a5c EXCE!Ordinal43+0x00002883 003e4a5c EXCE!Ordinal43+0x00004a5c 003e145 KERKE12!BaseThreadInitThunk+0x00000024 77719b63 ntdl!1_RtlUserThreadStart+0x000001b	
5a2db58a msc99Lwin32client+0x0013b58a 0085ea9f EXCEL!Ordinal43+0x0047ea9f 0050de55 EXCEL!Ordinal43+0x00050b57 00430b57 EXCEL!Ordinal43+0x00050b57 00430b56 EXCEL!Ordinal43+0x000509dc 00466fb9 EXCEL!Ordinal43+0x00086fb9 0040888 EXCEL!Ordinal43+0x00084fb9 0040888 EXCEL!Ordinal43+0x00004a5c 003e1194 EXCEL!Ordinal43+0x00004a5c 003e1194 EXCEL!Ordinal43+0x0000194 76380e34 KERKEL32!BaseThreadIntThunk+0x00000024 77719bc3 ntdl!!_RtlUserThreadStart+0x000001b	
0085ea9f EXCEL10rdinal43+0x0047ea9f 0050de55 EXCEL10rdinal43+0x0012de55 00430b57 EXCEL10rdinal43+0x00050b57 00430b5f EXCEL10rdinal43+0x000509dc 00466fb9 EXCEL10rdinal43+0x00086fb9 00468fb83 EXCEL10rdinal43+0x0002883 003e4a5c EXCEL10rdinal43+0x00004a5c 003e4194 EXCEL10rdinal43+0x00001194 76388e94 KERNE1321BaseThreadInitThunk+0x00000024 77719bc3 ntdl11Rt1UserThreadStart+0x000001b	
00430b97 EXCEL!Ordinal43+0x00050b97 00430b5f EXCEL!Ordinal43+0x00050b5f 004309dc EXCEL!Ordinal43+0x000509dc 00466fb9 EXCEL!Ordinal43+0x00028083 003e4a5c EXCEL!Ordinal43+0x00004a5c 003e4194 EXCEL!Ordinal43+0x00001194 76388e94 KERNEL32!BaseThreadInitThunk+0x00000024 77719bc3 ntdll!_RtlUserThreadStart+0x000002b 77719b92 ntdll!_RtlUserThreadStart+0x000001b	
004309dc EXCEL!Ordinal43+0x000509dc 00466fb EXCEL!Ordinal43+0x00086fb9 00408883 EXCEL!Ordinal43+0x00028883 003e4a5c EXCEL!Ordinal43+0x00004a5c 003e194 EXCEL!Ordinal43+0x00001194 76388e94 KERNEL32!BaseThreadInitThunk+0x00000024 77719b63 ntdll!_RtlUserThreadStart+0x0000002b 77719b92 ntdll!_RtlUserThreadStart+0x000001b	
00466fb9 EXCEL!Ordinal43+0x00086fb9 0040883 EXCEL!Ordinal43+0x0002883 003e4a5c EXCEL!Ordinal43+0x00004a5c 003e194 EXCEL!Ordinal43+0x00001194 76388e94 KERNEL32!BaseThreadInitThunk+0x00000024 77719bc3 ntdl!!_RtlUserThreadStart+0x0000002b 77719b92 ntdl!!_RtlUserThreadStart+0x0000001b	
003e4a5c EXCEL!Ordinal43+0x00004a5c 003e194 EXCEL!Ordinal43+0x00001194 76388e94 KERNEI32 BaseThreadInitThunk+0x00000024 77719bc3 ntdll!_RtlUserThreadStart+0x0000002b 77719b92 ntdll!_RtlUserThreadStart+0x0000001b	
76388e94 KERNEL32!BaseThreadInitThunk+0x00000024 77719bc3 ntdll!RtlUserThreadStart+0x000002b 77719b92 ntdll!_RtlUserThreadStart+0x000001b	
77719bc3 ntdll!RtlUserThreadStart+0x0000002b 77719b92 ntdll!_RtlUserThreadStart+0x0000001b	
0:000> k	
# ChildEBP RetAddr	
<u>00</u> 00306d3c 7431e5f5 gdi32full!FindMatchingPair+0x30 <u>01</u> 0030711c 7430be46 gdi32full!UnicodeBidiAlgorithm+0x925	
01 0030711c 7430be46 gdi32full/DnicodeBidiAlgorithm+0x925 02 00307164 74326d05 gdi32full/ScriptItemizeCommon+0xb6 02 00307184 5a2565bf gdi32full/ScriptItemize+0x35	
WARNING: Stack unwind information not available. Following frames may be wrong.	
04 0030735c 5a256a9f mso99Lwin32client+0xb65bf 05 003073e8 5a2da51d mso99Lwin32client+0xb6a9f	
06 0030748c 5a2db58a msc99Lwin32client+0x13a51d 07 003075c 009Eradf msc99Lwin32client+0x13b58a	
07 0030756c 0085ea9f mso99Lvin32client+0x13b50a 00 0030b6c8 0050de55 EXCEL Ordinal43+0x47ea9f 09 0030b714 00430b97 EXCEL Ordinal43+0x12de55	
U9 0030b/14 00430b9/ EXCELIOrdinal43+Us12de55 Qa 0030b/30 00430b9/ EXCELIOrdinal43+Us20b97	
0b 0030b764 004309dc EXCEL!Ordinal43+0x50b5f 0c 0030f81c 00466fb9 EXCEL!Ordinal43+0x509dc	
0d 0030f8f4 00408883 EXCELLOrdinal43+0x86fb9	
0e 0030f9dc 003e445c EXCEL Ordinal43+0x28883 0f 0030fbc8 003e1194 EXCEL Ordinal43+0x485c 10 0030fc14 75388e94 EXCEL Ordinal43+0x1194	
10 0030fc14 76388e94 EXCEL/Ordinal43+0x1194	
11 0030fc28 77719bc3 KERNEJ32/BaseThreadInitThunk+0x24 12 0030fc70 77719b92 ntdllRtUSerThreadStart+0x2b	
13 0030fc80 00000000 ntdll!_RtlUserThreadStart+0x1b 0:000> lmv m excel	
Browse full module list	
start end module name 003e0000 01db80000 <u>EXCEL</u> (export symbols) C:\Program Files\Microsoft Office\Office16\EXCEL.EXE	
Loaded symbol image file: C:\Program Files\Microsoft Office\Office16\EXCEL.EXE Image path: C:\Program Files\Microsoft Office\Office16\EXCEL.EXE	
Image name: EXCEL.EXE	
<u>Brovse all global symbols functions data</u> Timestamp: Thu Jul 30 05:05:36 2015 (55BA1310)	
CheckSum: 019DE26A ImageSize: 019D8000	
File version: 16.0.4266.1001	
Product version: 16.0.4266.0 File flags: 22 (Mask 3F) Pre-release Special	
File OS: 40004 NT Win32 File type: 1.0 App	
File date: 00000000.00000000	
Translations: 0000.04e4 CompanyName: Microsoft Corporation	
ProductName: Microsoft Office 2016 InternalName: Excel	
OriginalFilename: Excel.exe	
ProductVersion: 16.0.4266.1001 FileDermins: 16.0.4266.1001	
FileVersion: 16.0.4266.1001	
FileVersion: 16.0.4266.1001 FileDescription: Microsoft Excel	
FileDescription: Microsoft Excel	
FileVersion: 16.0.4266.1001 FileDescription: Microsoft Excel <	>

Figure 3: windbg output of Out-of-Bound SCRIPT_ITEM buffer dereference

In conclusion, the root-cause of this bug is the insufficiently-sized *pltems* buffer that Excel (or Mso99Lwin32client) allocates for ScriptItemize(). This is probably due to the developers' oversight when



. . .

reading the documentation for the ScriptItemize() function. From MSDN¹, the *cMaxItems* and *pItems* parameters are described as such:

clnChars [in]
 Number of characters in pwclnChars to itemize.
cMaxItems [in]
 Maximum number of SCRIPT_ITEM structures defining items to process.
...
pltems [out]

Pointer to a buffer in which the function retrieves **SCRIPT_ITEM** structures representing the items that have been processed. The buffer should be (cMaxItems + 1) * sizeof (SCRIPT_ITEM) bytes in length. It is invalid to call this function with a buffer to hold less than two **SCRIPT_ITEM** structures. The function always adds a terminal item to the item analysis array so that the length of the item with zero-based index "i" is always available as: pItems[i+1].iCharPos - pItems[i].iCharPos;

The developers probably noted that since *cMaxItems* is the maximum number of SCRIPT_ITEM structures to process, they allocated the *pItems* buffer to only *cMaxItems* structures where in fact, it should have *cMaxItems*+1.

On 12 Sept 2017, Microsoft identified this vulnerability as a "Uniscribe Remote Code Execution Vulnerability", and listed Windows 8, Windows 10, Windows Server 2012 and Windows Server 2016 as affected products. So this vulnerability should have been patched in gdi32.dll (at the root-cause) instead of Excel.

¹ ScriptItemize function: https://msdn.microsoft.com/en-us/library/windows/desktop/dd368556(v=vs.85).aspx



Detailed Timeline

Date	Summary
2017-05-22	MWR Labs reported vulnerability and POC to MSRC
2017-05-22	MSRC acknowledged and opened case 38823
2017-05-23	MSRC responded that the team could not reproduce the issue
2017-05-23	MWR Labs sent crash dump to MSRC
2017-08-04	MSRC responded that this will be patched in September 2017
2017-09-12	MSRC assigned CVE-2017-8692 and released patch for this vulnerability
2017-11-23	MWR Labs released advisory