Microsoft Office CTaskSymbol Use-After-Free Vulnerability

17/08/2015

Software:	Microsoft Office
Affected Versions:	MS Office 2013 SP1 (x86, x64)
CVE Reference:	CVE-2015-1642
Author:	Yong Chuan, Koh (@yongchuank)
Severity:	Important
Vendor:	Microsoft
Vendor Response:	Fix released as part of MS15-081

Description

Microsoft Office is a suite of desktop applications consisting of Word, Powerpoint, Excel, Outlook and various other productivity applications. The applications are affected by a use-after-free vulnerability while parsing a specially crafted Office file as a result of the application loading the `CTaskSymbol` COM object in memory.

Impact

If persuaded to open the crafted Office file, a successful exploitation would allow an attacker to run arbitrary code in the context of the target application.

Cause

The vulnerability exists because Microsoft Office incorrectly dereferences the `CTaskSymbol` object after it is freed.

Interim Workaround

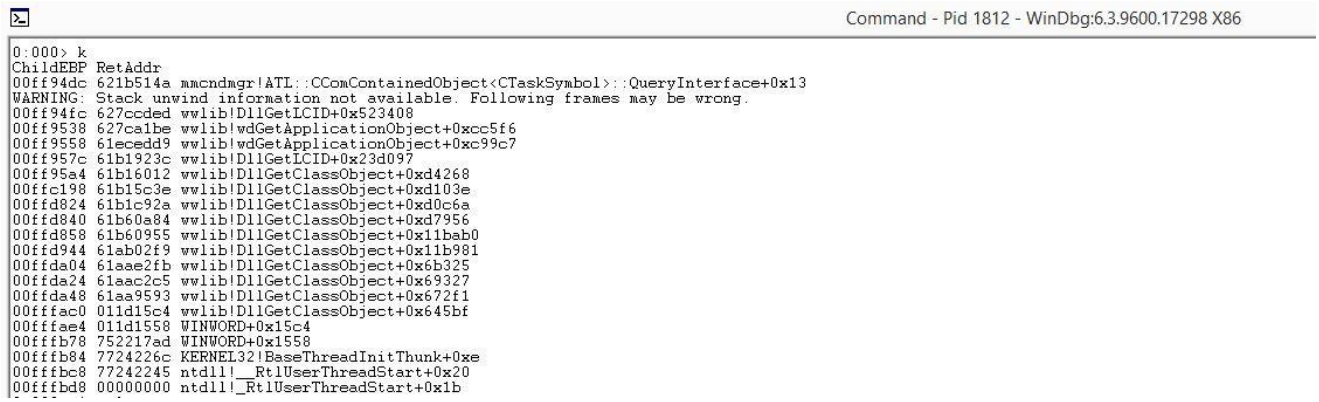
Avoid opening Office files from untrusted sources or view them in Protected-View mode.

Solution

Users should apply MS15-081 updates from Microsoft.

Technical Details

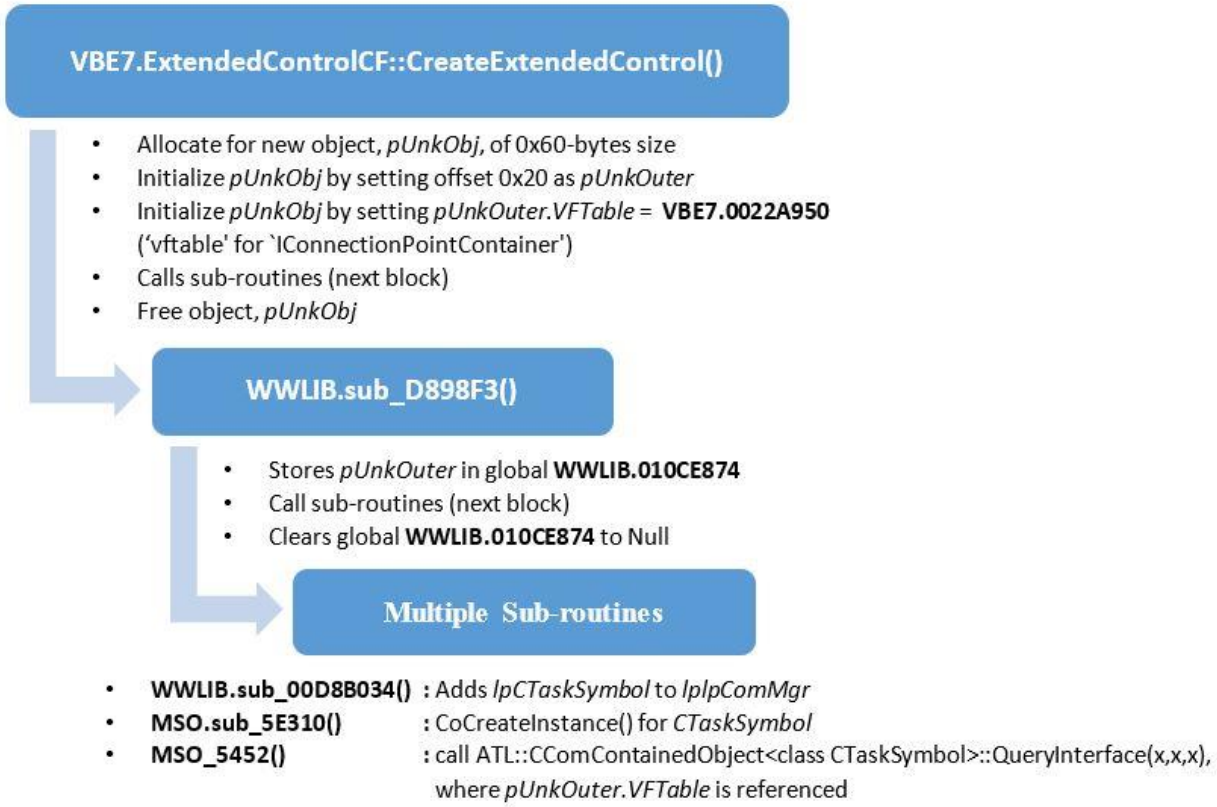
As WINWORD tries to load the `CTaskSymbol` COM (`UUID: 44F9A03B-A3EC-4F3B-9364-08E0007F21DF`), with page-heap enabled, it crashes at the following address:



In the first screenshot, the invalid `VFTable` read at address `0x59bc8c4b` occurs because the object has already been freed. With some reverse-engineering, it was found that this `VFTable` belongs to the aggregate object's `IUnknown` interface, `pUnkOuter`, of the `CTaskSymbol` object:

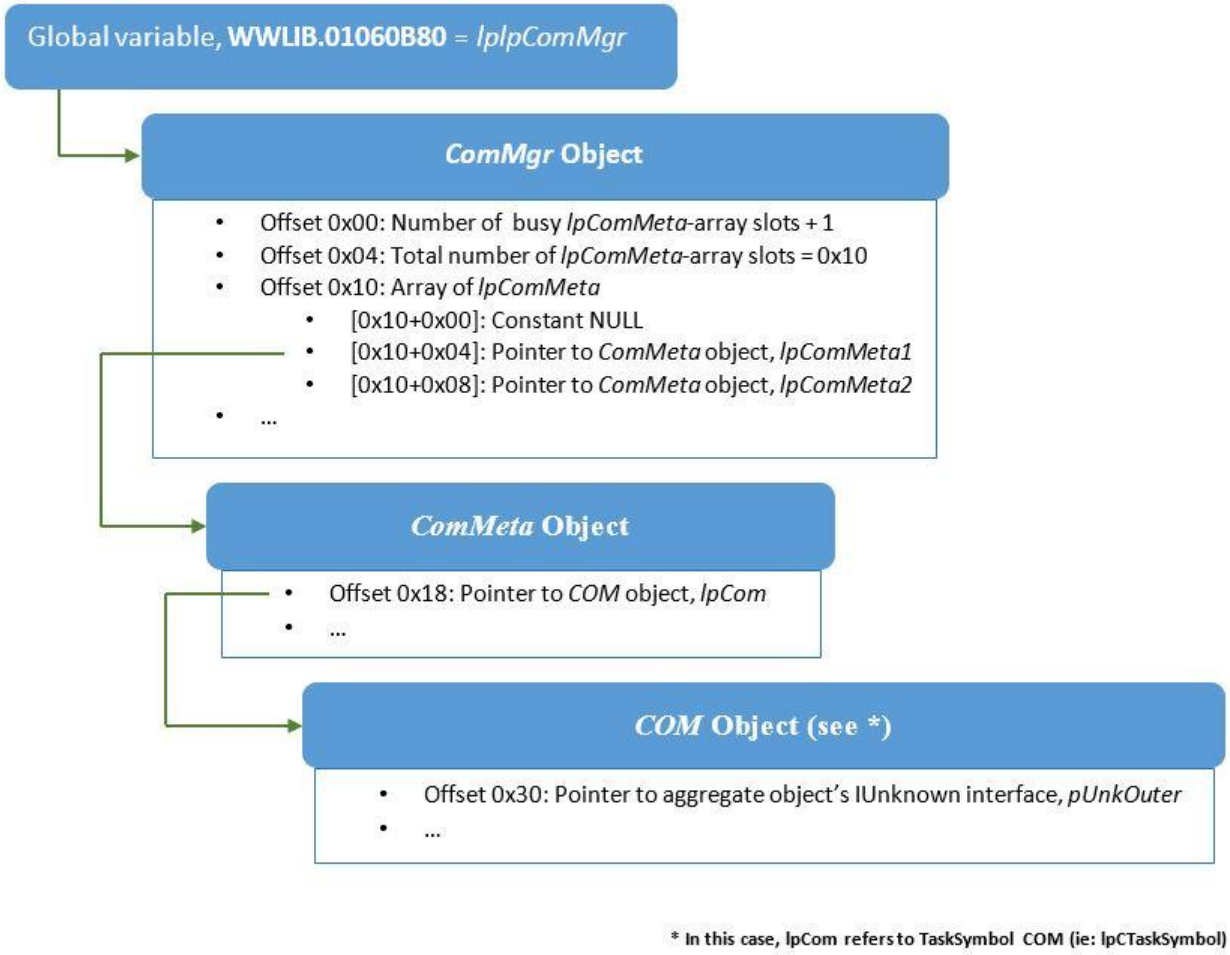
```
CoCreateInstance (
            rclsid       = {44F9A03B-A3EC-4F3B-9364-08E0007F21DF},
            pUnkOuter    = aggregate object's IUnknown interface,
            dwClsContext = CLSCTX_INPROC_SERVER|CLSCTX_INPROC_HANDLER,
            riid         = {00000000-0000-0000-C000-000000000046},
            ppv          = lpOutput
            )
```

This `pUnkOuter IUnknown` interface object is allocated, used and freed in the following sequence:



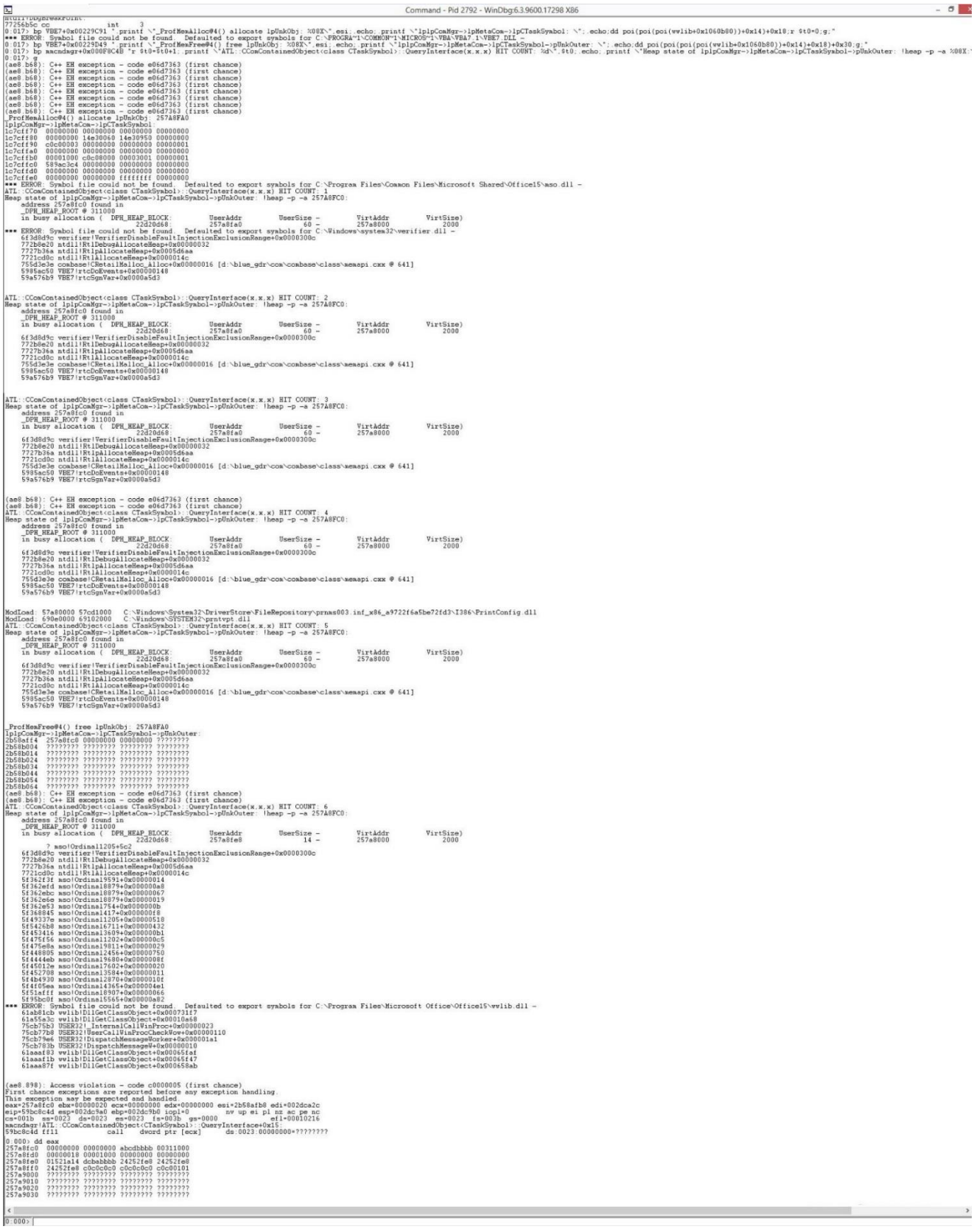
The series of actions to load a COM object starts in `VBE7.ExtendedControlCF::CreateExtendedControl()` where it allocates a new 0x60 bytes `pUnkObj` object, sets `pUnkOuter` to its +0x20 offset, calls into `WWLIB.sub_D898F3()` and finally frees `pUnkObj` . In `WWLIB.sub_D898F3()`, the target `pUnkOuter` is first stored in global `WWLIB.010CE874`, calls into some subroutines and finally clears the reference in `WWLIB.010CE874`. Among the subroutines, the `CTaskSymbol` COM is instantiated with `pUnkOuter` and then added to the internal COM manager, `lplpComMgr`.

As its name implies, the `lplpComMgr` COM manager help WINWORD tracks all COM objects that are embedded into the current document. Its structure and relationship to `pUnkOuter` of the `CTaskSymbol` COM is illustrated:



As the diagram above shows, `lplpComMgr` maintains an array of references to all embedded COM objects, whose details are described in each `lpCom` object. Among the details is the COM's aggregate object's `IUnknown` interface, `pUnkOuter`, at offset +0x30.

At this point, there are 2 references to `pUnkOuter`, one in global `WWLIB.010CE874` and the other in `lplpComMgr`. As WINWORD exits `VBE7.ExtendedControlCF::CreateExtendedControl()`, the `pUnkOuter` is freed (as part of `pUnkObj`) and its reference in `WWLIB.010CE874` is cleared. However it does not remove the reference in `lplpComMgr -> lpComMeta -> lpCTaskSymbol -> pUnkOuter`. As a result, in a subsequent call to `ATL::CComContainedObject<class CTaskSymbol>::QueryInterface(x,x,x)`, the invalid/freed `lpCTaskSymbol -> pUnkOuter -> VFTable` is being referenced. The following screenshot illustrates these:


Command - Pid 2792 - WinDbg:6.3.9600.17298 X86

Detailed Timeline

Date:	Summary:
27/02/2015	MWR Labs reported issue to Microsoft
28/02/2015	Microsoft acknowledge receipt and start investigation of issue
02/04/2015	Microsoft replied that issue is assigned CVE-2015-1642
11/08/2015	Microsoft released patch as part of MS15-081
17/08/2015	MWR Labs released advisory