TALOS-2016-0182

SYMANTEC NORTON SECURITY IDSVIX86 PE REMOTE SYSTEM DENIAL OF SERVICE VULNERABILITY

JULY 7, 2016

REPORT ID

CVE-2016-5308

SUMMARY

A denial of service vulnerability exists in the Portable Executable file scanning functionality of Symantec Norton Security. A specially crafted PE file can cause an access violation in IDSvix86 kernel driver resulting in denial of service. An attacker can trigger this vulnerability for example by emailing the victim the forged file.

TESTED VERSIONS

Symantec Corporation Norton Security 22.6.0.142, IDSvix86 driver version 15.1.0.1263

PRODUCT URLS

http://norton.com

CVSSV3 SCORE

7.5 - CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:L/I:N/A:H

DETAILS

This vulnerability occurs when Norton is trying to parse specifically crafted file Portable Executable file. The faulting code is located in the IDSvix86 driver:

```
...
caller:
.text:00047965                  push    edi             ; Section Raw Data
.text:00047966                  lea     ecx, [ebp+arg0_StackBuff]
.text:00047969                  push    ecx             ; StackBuff
.text:0004796A                  call    BugProc
BugProc:
.text:00058DC2                  push    ebp
.text:00058DC3                  mov     ebp, esp
.text:00058DC5                  push    ebx
.text:00058DC6                  push    esi
.text:00058DC7                  mov     esi, [ebp+arg0_StackBuff]
.text:00058DCA                  mov     ecx, [esi+10h]  ; ecx=0
.text:00058DCD                  mov     eax, ecx
.text:00058DCF                  push    edi
.text:00058DD0                  mov     edi, [ebp+SectionRawSize] ; edi=raw size
...
.text:00058E26 loop_until_all_data:             ; CODE XREF: BugProc+70j
.text:00058E26                  mov     edx, [ebp+arg4_SectionRawData]
.text:00058E29                  lea     eax, [edx-ebx-3Fh]
.text:00058E2D                  push    eax             ; read buff (raw section data
 + offset)
.text:00058E2E                  push    esi             ; store buff (stack)
.text:00058E2F                  call    MD5Compress
.text:00058E34                  add     [ebp+arg0_StackBuff], 40h
.text:00058E38                  add     ebx, 40h
.text:00058E3B                  cmp     ebx, edi
.text:00058E3D                  jb      short loop_until_all_data ; until raw size
---
```

Loop at 0x00058E26 is executed till the counter ebx is less than section's raw size (comparison at 0x00058E3B) which is controlled freely by the attacker. If the SectionRawData parameter is big enough it can cause the MD5Compress function to access memory which is unavailable causing the machine to crash.

CRASH INFORMATION

```
...
kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

PAGE_FAULT_IN_NONPAGED_AREA (50)
Invalid system memory was referenced.  This cannot be protected by try-except,
it must be protected by a Probe.  Typically the address is just plain bad or it
is pointing at freed memory.
Arguments:
Arg1: a5fa9003, memory referenced.
Arg2: 00000000, value 0 = read operation, 1 = write operation.
Arg3: 8cd55713, If non-zero, the instruction address which referenced the bad memory
      address.
Arg4: 00000000, (reserved)

Debugging Details:
------------------

READ_ADDRESS:  a5fa9003 Paged pool
FAULTING_IP:
IDSvix86+48713
8cd55713 0fb67001        movzx   esi,byte ptr [eax+1]
MM_INTERNAL_CODE:  0
IMAGE_NAME:  IDSvix86.sys
DEBUG_FLR_IMAGE_TIMESTAMP:  5723ac68
MODULE_NAME: IDSvix86
FAULTING_MODULE: 8c0d4000 IDSvix86
DEFAULT_BUCKET_ID:  WIN7_DRIVER_FAULT
BUGCHECK_STR:  0x50
PROCESS_NAME:  svchost.exe
CURRENT_IRQL:  2
ANALYSIS_VERSION: 6.3.9600.17298 (debuggers(dbg).141024-1500) amd64fre
TRAP_FRAME:  922ab500 -- (.trap 0xffffffff922ab500)
ErrCode = 00000000
eax=a5fa9002 ebx=00000000 ecx=922ab590 edx=0000000c esi=00000000 edi=922ab60c
eip=8cd55713 esp=922ab574 ebp=922ab5c4 iopl=0         nv up ei pl nz na pe nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010206
IDSvix86+0x48713:
8cd55713 0fb67001        movzx   esi,byte ptr [eax+1]      ds:0023:a5fa9003=??
Resetting default scope
LAST_CONTROL_TRANSFER:  from 8291c083 to 828b8110

STACK_TEXT:
922ab84c 8291c083 00000003 43e8c867 00000065 nt!RtlpBreakWithStatusInstruction
922ab89c 8291cb81 00000003 845d1d68 0000598a nt!KiBugCheckDebugBreak+0x1c
922ab460 828cb41b 00000050 a5fa9003 00000000 nt!KeBugCheck2+0x68b
922ab4e8 8287e3d8 00000000 a5fa9003 00000000 nt!MmAccessFault+0x106
922ab4e8 8cd55713 00000000 a5fa9003 00000000 nt!KiTrap0E+0xdc
WARNING: Stack unwind information not available. Following frames may be wrong.
922ab5c4 8cd55e34 922ab60c a5fa8ff0 a5f963f0 IDSvix86+0x48713
922ab5e0 8cd4496f 00012c00 a5f963f0 ffffffff IDSvix86+0x48e34
922ab668 8cd455e1 922ab6c0 88394008 922ab694 IDSvix86+0x3796f
922ab678 8cd458f9 922ab6c0 88e61250 00000000 IDSvix86+0x385e1
922ab694 8cd45737 922ab6c0 88e59538 88e61198 IDSvix86+0x388f9
922abed4 8752579c a3a065a8 40000002 00000002 IDSvix86+0x18737
922ab728 8752980f a3a065a8 88e61198 a30e7f58 SYMEFASI+0x11179c
922ab75c 87528e79 88e61198 922ab7a4 a3a71008 SYMEFASI+0x11580f
922ab774 9c8a7cd4 00000002 922ab798 00000002 SYMEFASI+0x1146f9
922ab7b0 9c8aa3f9 828ff8f7 a3a7100c a3a71008 SRTSP+0x95cd4
922ab7f8 9c8aa6c8 a3a71008 9658fbd0 0b0899ec SRTSP+0x983f9
922ab818 9c865e49 922ac000 0b0899ec 8287461 SRTSP+0x986c8
922ab834 9c8660c5 922ab88c 9658fbd0 62ca0002 SRTSP+0x53e49
922ab854 9c83ed44 847db748 847db800 847db7a8 SRTSP+0x540c5
922ab868 86e93324 847db7a8 9658fbd0 00000000 SRTSP+0x2cd44
922ab8d0 86e96512 007db748 847db748 1000000c fltmgr!FltpPerformPostCallbacks+0x24a
922ab8e4 86e96b46 847db748 8478c618 922ab924 fltmgr!FltpProcessIoCompletion+0x10
922ab8f4 86e9759c 856127a8 8478c618 847db748 fltmgr!FltpPassThroughCompletion+0x98
922ab924 86eaa8c9 922ab944 00000000 00000000 fltmgr!FltpLegacyProcessingAfterPreCallbacksCo
mpleted+0x33a
922ab970 82874593 856127a8 8562e008 8453ef24 fltmgr!FltpCreate+0x2db
922ab988 82a842a9 43e8c290 922abb30 00000000 nt!IofCallDriver+0x63
922abb60 82a63ac5 855d6e20 853aed20 84734498 nt!IopParseDevice+0xed7
922abadc 82a73ed6 00000000 922abb30 00000040 nt!ObpLookupObjectName+0x4fa
922abb38 82a6a9b4 010eedd6 843aed20 00000001 nt!ObOpenObjectByName+0x165
922abbb4 82a8e218 010eee30 80100080 010eedd4 nt!IopCreateFile+0x673
922abc00 8287b1aa 010eee30 80100080 010eedd4 nt!NtCreateFile+0x34
922abc00 77a370b4 010eee30 80100080 010eedd4 nt!KiFastCallEntry+0x12a
010eed90 77a355d4 75dcaa21 010eee30 80100080 ntdll!KiFastSystemCallRet
010eed94 75dcaa21 010eee30 80100080 010eedd4 ntdll!NtCreateFile+0xc
010eee38 75dcca9c 00000060 80100080 00000003 KERNELBASE!CreateFileW+0x35e
010eefb4 75dcbb5d 010eef40 010eefd8 00000020 KERNELBASE!BasepLoadLibraryAsDataFileInternal
+0x280
010eefec 75a488c4 01a9d270 00000000 00000001 KERNELBASE!LoadLibraryExW+0xf6
010ef030 75a4888b 00000001 01a9d270 00008001 apphelp!GetFileVersionInfoSizeExW+0x30
010ef044 75a487d2 01a9d270 010ef068 2137837f apphelp!GetFileVersionInfoSizeW+0x12
010ef094 75a455d5 02a123c0 01a9d080 01a9d140 apphelp!SdbpGetVersionAttributes+0xdd
010ef0a8 75a4556c 02a123c0 01a9d080 00006014 apphelp!SdbpGetAttribute+0xa9
010ef0dc 75a45476 02a123c0 01a9d080 00006014 apphelp!SdbpCheckAttribute+0xaf
010ef10c 75a4538f 02a123c0 01af4b98 00033160 apphelp!SdbpCheckAllAttributes+0xa1
010ef3bc 75a45064 02a123c0 01af4b98 00033198 apphelp!SdbpCheckForMatch+0x560
010ef3f4 75a457a0 02a123c0 00000000 00033198 apphelp!SdbpCheckExe+0x1c1
010ef470 75a44cc8 02a123c0 01af4b98 00007007 apphelp!SdbpSearchDB+0xa2
010ef7b0 75a42f2d 02a123c0 00153080 00000000 apphelp!SdbGetMatchingExeEx+0x3ee
010ef7e4 75a431ca 000001bc 00000000 00000000 apphelp!InternalCheckRunApp+0x2eb
010efab8 730113b7 000001bc 00000000 00000000 apphelp!ApphelpCheckRunAppEx+0xed
010efb7c 730115bf 01100808 01b09058 010efbb0 aelupsvc!AelpProcessCacheExeMessage+0x20d
010efb8c 77a02661 010efbec 011008c0 01b09058 aelupsvc!AelTppWorkCallback+0x19
010efbbd 77a20842 0106fbec 01b090b8 76a2c554 ntdll!TppWorkpExecuteCallback+0x10f
010efd10 76773c45 01b0c500 010efd5c 77a537f5 ntdll!TppWorkerThread+0x572
010efd1c 77a537f5 01b0c500 76a2c518 00000000 kernel32!BaseThreadInitThunk+0xe
010efd5c 77a537c8 77a203e7 01b0c500 00000000 ntdll!__RtlUserThreadStart+0x70
010efd74 00000000 77a203e7 01b0c500 00000000 ntdll!_RtlUserThreadStart+0x1b

STACK_COMMAND:  kb
FOLLOWUP_IP:
IDSvix86+48713
8cd55713 0fb67001        movzx   esi,byte ptr [eax+1]
SYMBOL_STACK_INDEX:  5
SYMBOL_NAME:  IDSvix86+48713
FOLLOWUP_NAME:  MachineOwner
FAILURE_BUCKET_ID:  0x50_IDSvix86+48713
BUCKET_ID:  0x50_IDSvix86+48713
ANALYSIS_SOURCE:  KM
FAILURE_ID_HASH_STRING:  km:0x50_idsvix86+48713
FAILURE_ID_HASH:  {3bd6bb7f-4849-a2c5-fcf6-1882fde5c02c}
Followup: MachineOwner
---------
---
```

CREDIT

Discovered by Piotr Bania of Cisco Talos.

TIMELINE

2016-05-31 - Vendor Notification
2016—07-07 – Patch Released
2016—07-07 – Public Disclosure