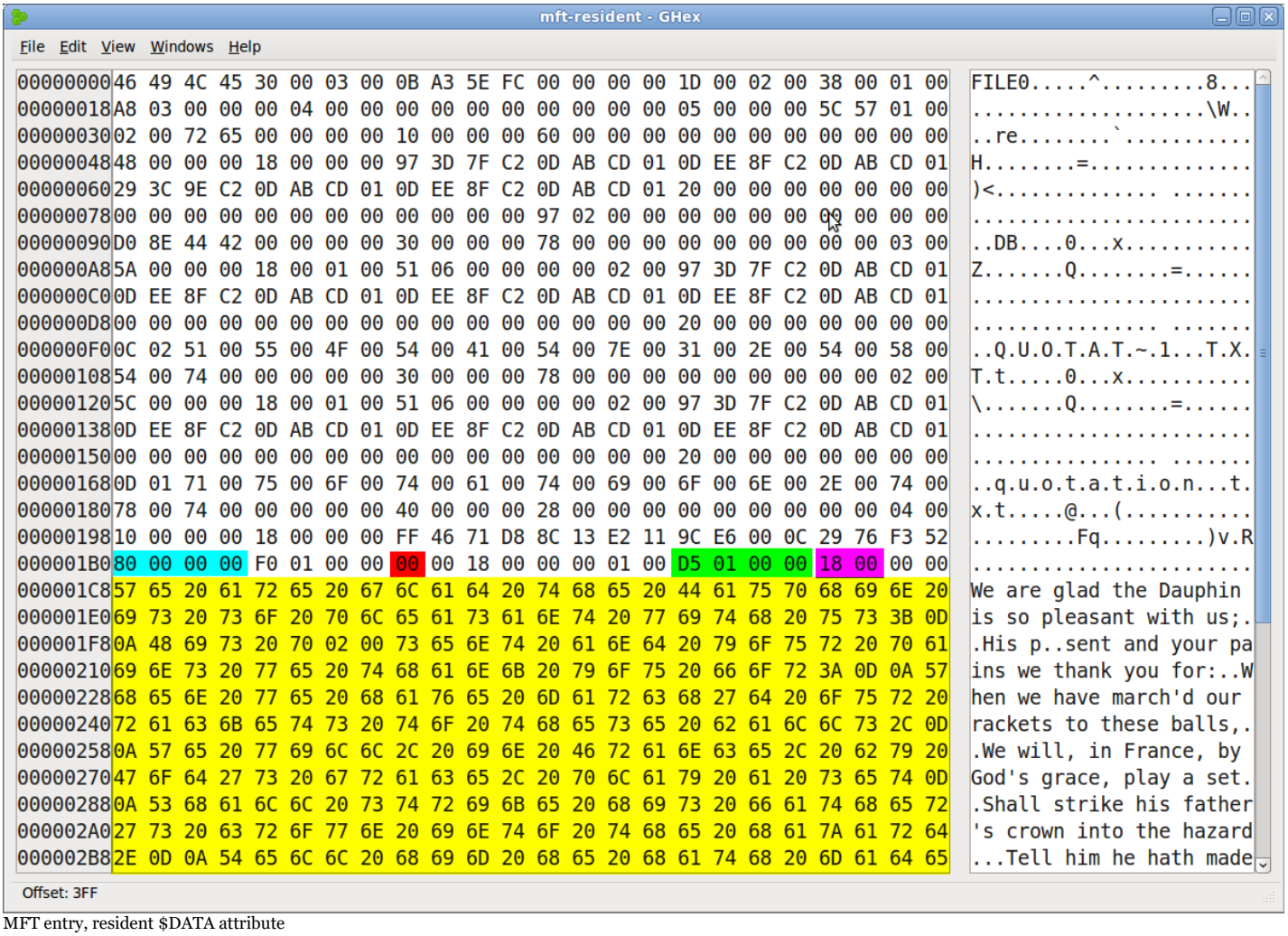
SANS Digital Forensics and Incident Response Blog | Resident $DATA Residue in NTFS MFT Entries

Hal Pomeranz, Deer Run Associates

I came across a small but interesting artifact in the course of a recent investigation. Quick Google searching failed to find any documentation elsewhere, so here's a brief summary of my findings. The bottom line is that residue of old resident $DATA entries may exist in NTFS MFT records after the file data has grown large enough to become a non-resident attribute.

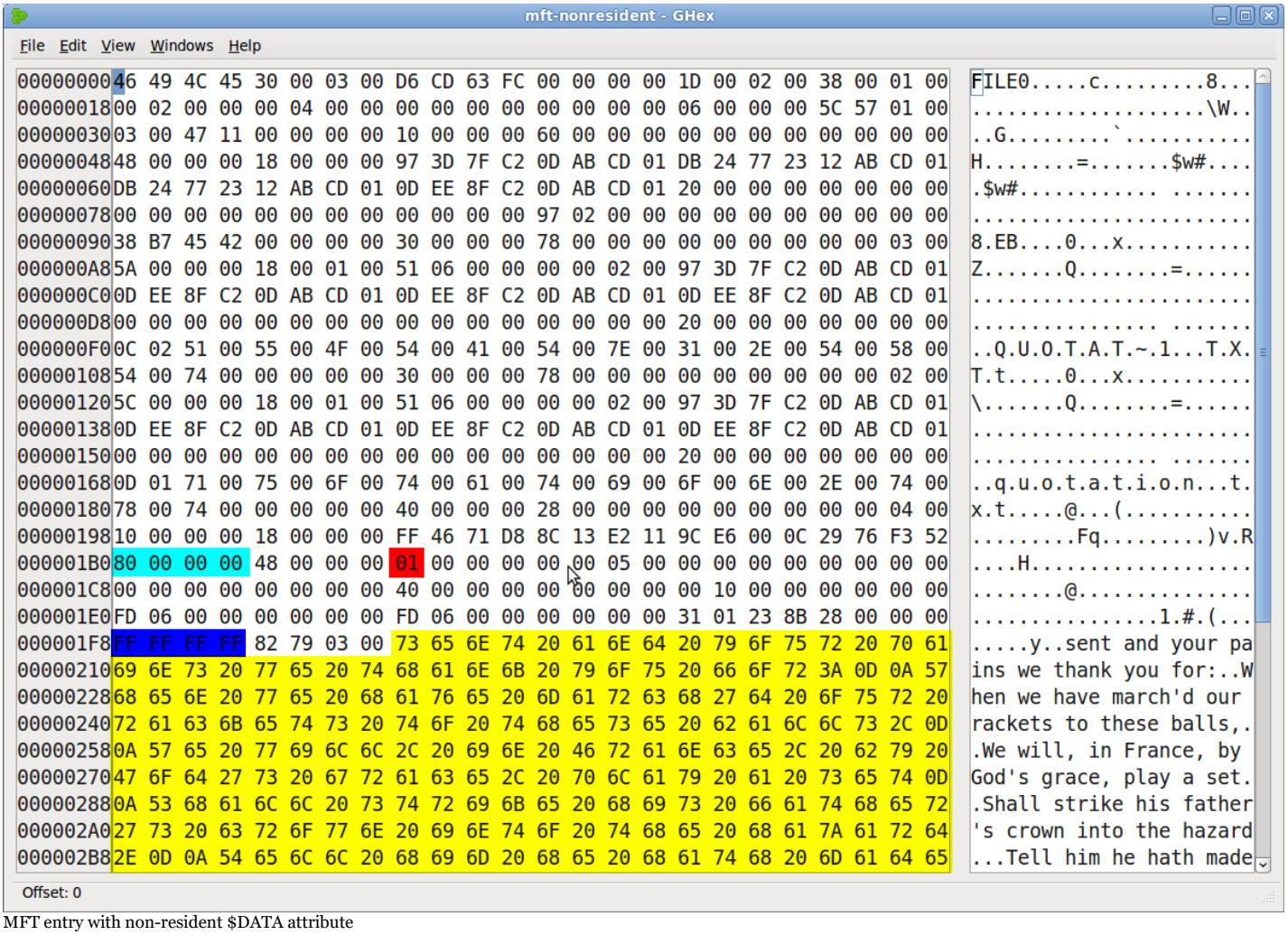
You can actually force this situation to occur yourself. First, create a small text file using Notepad or your preferred text editor (Carrier suggests limiting the file size to less than 700 bytes if you want the file to be stored as a resident attribute). Here's a picture of the MFT entry for the file I created:



MFT entry, resident $DATA attribute

The $DATA attribute (0x80) starts at byte offset 432 (0x01B0). The non-resident flag is zero, meaning this is a resident $DATA attribute. The size of the resident data is 469 bytes (0x01D5) starting 24 bytes (0x18) from the beginning of the attribute. You can see the text starting at byte offset 456 (0x01C8) in the hex editor.

The interesting stuff happens when you add more data to your file. In my case, I added a little over 1K additional data in my Notepad buffer to force the attribute to become non-resident. I dumped out the MFT entry again, and here's the hex editor view:



MFT entry with non-resident $DATA attribute

The $DATA attribute still starts at byte offset 432. In the updated entry, however, the non-resident flag is 0x01, indicating that the $DATA attribute is non-resident. I'm not going to bother decoding the rest of the $DATA attribute because you can quite clearly see the 0xFFFFFFFF end of file marker that marks the end of the new MFT entry (byte offset 504).

The four bytes after the end of file marker are also modified. These four bytes take us to the end of the sector boundary. From that point forward, you can see the contents of the original resident $DATA. While it doesn't show in the images, the last 512 bytes of both MFT entries are the same.

This residual data gives us a fragment of the contents of a file at a particular point in time. This version of the file may have only existed for a brief period and not been captured in a Volume Shadow Copy or other backup. In my investigation I got a string hit on the residual data in the MFT entry while the current (and non-resident) version of the file did not contain the string of interest. Both versions ended up being relevant to the investigation, but the historical relic of the residual data made the combined find even more interesting.

Hal Pomeranz is an Independent Digital Investigator, a SANS Institute Faculty Fellow, and a GCFA. He doesn't always analyze NTFS, but when he does he often uses a hex editor. Hal will be teaching For508: Advanced Computer Forensic Analysis and Incident Response in San Antonio, Nov 27 - Dec 1.