Expose Evidence of Timestomping with the NTFS Timestamp Mismatch Artifact

By Rick Andrade

Published: 2020-08-24 · Archived: 2026-04-06 01:55:45 UTC

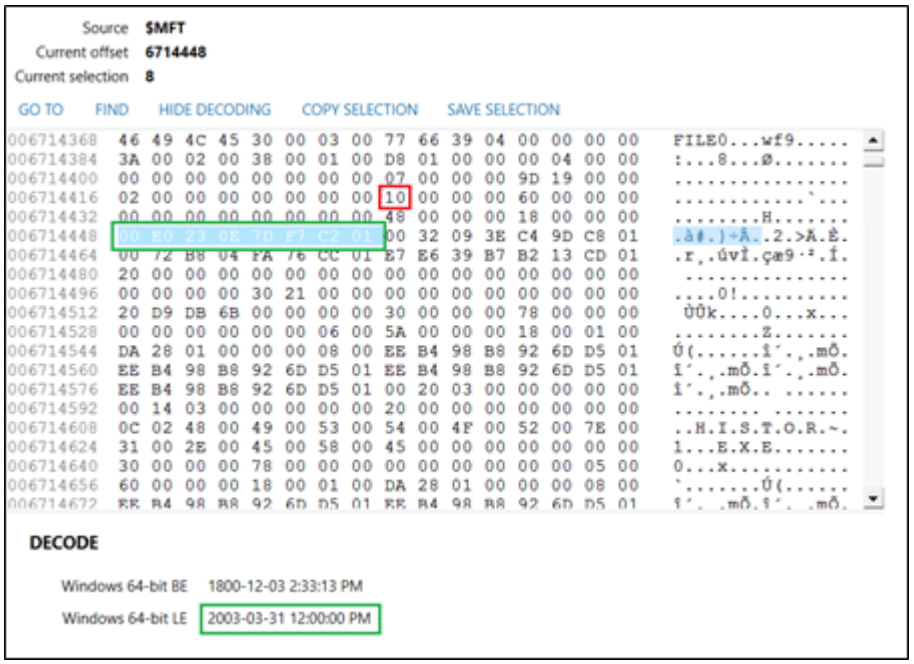
Malicious activity can devastate the infrastructure it infects, and so it is increasingly important to be able to first identify suspicious behavior so that you can begin remediating its affects. Unfortunately, the goal of malware is to blend in, go unnoticed, and hide from its target so that it can maintain its presence on the target endpoint. One potential way that some malicious actors try to accomplish this task is to manipulate the timestamps of the malicious file(s), a tactic known as timestomping.

The goal of timestomping is to edit the timestamps being displayed and reported to the end user and incident responders in an attempt to make it seem as though the file doesn't fall into the timeline of other detected malicious activity. When the incident responder starts reviewing alerts, logs, and other artifacts from the infected machine, a timestomped file might fall outside of the scope of investigation if the timestamps are maliciously manipulated. The result could be an undetected malicious file that can persist on the infected endpoint.

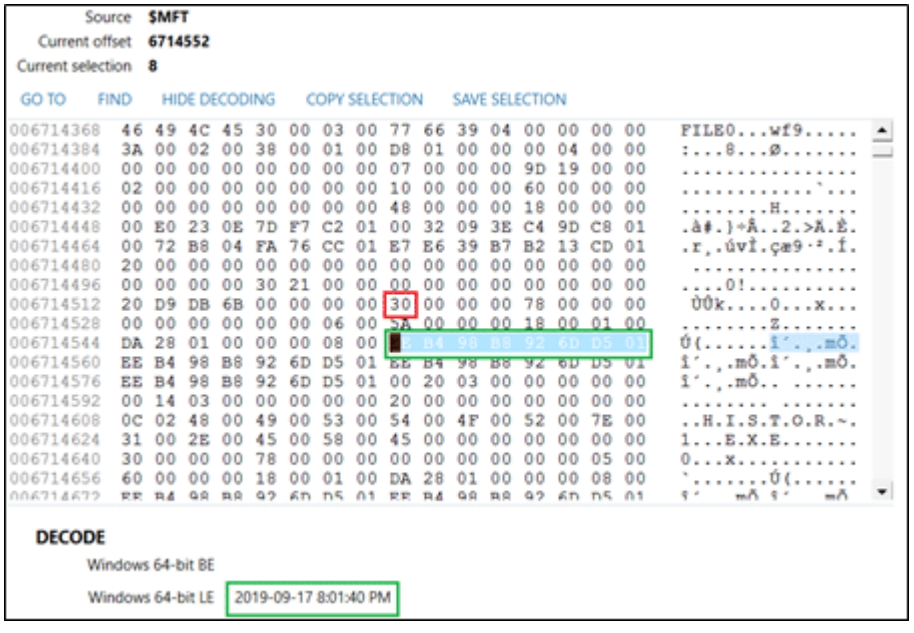
Often, though, this activity can be detected by comparing multiple timestamps associated within the MFT record corresponding to the file in question. The NTFS Timestamp Mismatch artifact, gives you a starting point in the incident response investigations in which you suspect timestomping may have occurred. Here is how it works!

Within an MFT record of a file stored within a NTFS endpoint, there are multiple sections, or attributes, that contain various types of information about a file. For this new artifact, we will be focusing on the $Standard_Information ($SI) and $File_Name ($FN) attributes. Both sections of the MFT record contain sets of timestamps: Created, Accessed, Modified, and MFT Modified.

The $SI section of the MFT record is indicated with the value 0x10, as outlined in red below, and the Created timestamp is highlighted and decoded as well in green. The $SI timestamps are what Windows would display the end user as well as what most forensic tools will display as far as dates/times stamps in the File System view.



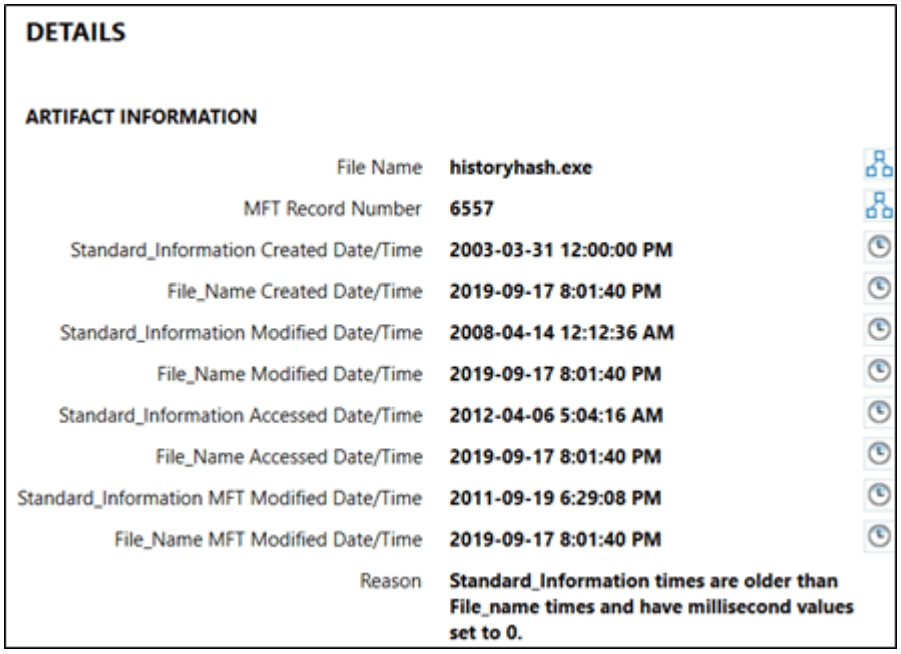
Outlined below in red, the $FN section is indicated with the value 0x30, and the Created timestamp is highlighted and decoded again in green as well. The $FN timestamps in the MFT record are only modified by the Windows kernel and will generally go untouched by antiforensic timestomping tools.



In the above example screenshots, the MFT record is from a timestamp manipulated file, and you can see that when the timestamps from both the $SI and $FN are decoded, the difference is worth noting.

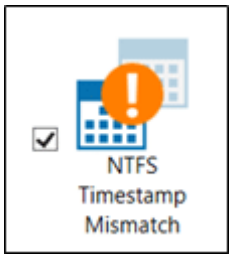
Now, in the NTFS Timestamp Mismatch artifact, AXIOM will automatically analyze both sets of timestamps for evidence of timestomping. Each artifact hit will give you both sets of timestamps, as well as a reason for the

artifact hit.



First, this artifact will compare the timestamps within the MFT Records of files in the file system from both the $SI and the $FN attributes, and will flag a mismatch when the $SI timestamp is earlier than the $FN timestamp. Additionally, this artifact will check to see if the millisecond values in the timestamp are exactly zero, which can also sometimes be a potential indicator that timestomping activity may have occurred on an infected system. For a positive hit on this artifact, only one of these criteria needs to be true, and the reason will be listed in the details panel in AXIOM Examine.

Keep in mind that this artifact is disabled by default in AXIOM Process, so be sure to select it when processing if you believe that timestamp manipulation may have occurred on your Windows endpoint.



This artifact can help provide you with a starting point if you believe timestomping activity occurred on an infected system and allow you to properly timeline activity on your infected endpoint alongside IDS alerts, network logs, and additional artifacts in your case. Note, however, that there could be legitimate reasons from normal system behavior that could cause this mismatch, as well as ways that malicious activity can circumvent this timestamp difference (for example, as referenced in this MITRE blog).
