Defence Evasion Technique: Timestomping Detection – NTFS Forensics

April 28, 2022

Forensic analysts are often taught two methods for detecting file timestomping that can lead to blind spots in an investigation. The two most well-taught methods for analysts to detect timestomping are:

- Compare the $STANDARD_INFORMATION timestamps vs the $FILE_NAME timestamps in the Master File Table (MFT)

- Look for nanoseconds in a timestamp matching "0000000" as this often shows the use of an automated tool (i.e. Metasploit)

These two detection methods are based on two fallacies that I will explore in this blog post:

- **Myth 1:** $FILE_NAME timestamps cannot be timestomped
- **Myth 2:** Attacker tools cannot alter nanoseconds in a timestamp

INTRODUCTION TO TIMESTOMPING

Timestomping is a technique where the timestamps of a file are modified for defence evasion. Threat actors often perform this technique to blend malicious files with legitimate files so that when an analyst is performing IR, critical evidence escapes detection.

Timestomping using tools like Cobalt Strike (offensive-security tool), Timestomp.exe (timestomping tool) and Metasploit (offensive-security framework) will result in timestamp changes to the MAC(b)

There are two attributes that record times in an MFT file – the $STANDARD_INFORMATION ($SI) and the $FILE_NAME ($FN) attribute. Each of these attributes stores the MAC(b) times for the file accordingly. For files with filenames that are longer, there will be two corresponding $FN MAC(b) attributes totalling another 8 timestamps on top of the existing $SI timestamps.

Modification of the $SI timestamp is the most common method of timestomping as it can be modified at the user-level using a set of API calls. However, modification of the $FN attribute requires the kernel – OR abuse of how the $FN timestamps are set (when a file is renamed or moved).

As such, there are two methods for modifying the $FN timestamps:

- **Method 1**
  Modification of $FN on older operating systems where Patch Guard hasn't been introduced using Windows API calls to the native APIs NtSetInformationFile and NtQueryInformationFile.

- **Method 2**
  Modification of $FN on any OS by timestomping the $SI attribute and then moving or renaming the file to copy the $SI time into the $FN time. This technique was discussed in this Black Hat Presentation.

Threat Actors Utilising Timestomping

Just to demonstrate for those who haven't come across timestomping in an investigation – threat actors and malware often use this technique. Some notable threat actors that have used timestomping during their attacks include (and is not limited to):

- IRON TWILIGHT (APT28)
- IRON HEMLOCK (APT29) – Timestomped their backdoors
- TIN WOODLAWN (APT32) – Timestomped raw XML scheduled task files and modified the CREATE times.
- NICKEL GLADSTONE (APT38) – Modifying file times to match other files on the system.
- NICKEL ACADEMY (Lazarus) – Copying timestamp from calc.exe to their malicious dropped files (this is something that Cobalt Strike does)

If you're interested in reading more about this - check out the Mitre attack page for this technique.

Patch Guard does not exist, you can use SetMace to alter the $FN timestamps by relying on the API calls (NTSetInformationFile and NTQueryInformationFile).

Step 1: Timestomp the $SI attributes by setting nanosecond precision

The tool I am using here is nTimetools and as you can see here, this already sets the nanosecond to an arbitrary attacker-defined number. This defeats the first detection that's commonly taught to "look for .0000000 in the nanoseconds". This has been documented by Forensics Wiki, Mari DeGrazia and Harlan Carvey.

When you analyse the MFT just to track the changes you can see that these are the timestamps for the $SI and $FN attributes.

$STANDARD_INFORMATION Timestamps:

Creation – 2020-05-19 12:34:56
Modification - 2020-05-19 12:34:56
MFT change – 2020-05-19 12:34:56
Last Access - 2020-05-19 12:34:56

$FILENAME Timestamps:

Creation – 2022-04-21 02:45:38
Modification - 2022-04-21 02:45:38
MFT change - 2022-04-21 02:55:01
Last Access - 2022-04-21 02:45:38

Step 2: Move the file

In this instance I moved my "file.txt" from my Desktop into the Temp folder.

Modification - 2020-05-19 12:34:56
MFT change – 2020-04-21 02:55:01
Last Access - 2020-05-19 12:34:56

$FILENAME Timestamps:
Creation – 2020-05-19 12:34:56
Modification - 2020-05-19 12:34:56
MFT change - 2020-05-19 12:34:56
Last Access - 2020-05-19 12:34:56

As you can see here, the $FN attributes have been changed. The threat actors at this point just need to timestomp the MFT change time and then you have a perfect set of timestamps.

Step 4: Alternate Method
If you're faced with an older version of windows without Patch Guard – you can use the SetMace tool and rely on API manipulation of the timestamps. I performed this on a 32-bit Windows 2003 Server and got the same results:

First I created a file on my Desktop named "test.txt"

Then proceeded to run SetMace to timestomp both $FI and $FN attributes:

And then I parsed out the $MFT and you can see here that the $FN and $SI attributes were both manipulated:

DETECTION METHODOLOGY

For most cases of timestomping where an attacker uses one of the following methods below - the detection methods of comparing $FI and $SI along with looking at nanosecond precision will suffice.

However, as demonstrated above, it's almost trivial to bypass these two detection mechanisms which will force an analyst to consider other methods for detection. The best method I have found for detecting these anomalies is by analysing the USN Journal file and the $Logfile. On busy systems, the $Logfile may be overwritten very quickly which would force an analyst to consider pulling a $Logfile from a shadow copy. However, just the USN Journal file will show enough information for an analyst to "question" the authenticity of the timestamps.

USN Journal Detection
As you can see in my parsed USN Journal output below, there are multiple operations that are tracked:

- FILE_CREATE
- RENAME_OLD_NAME
- RENAME_NEW_NAME

For all these timestamps that are tracked, you can see the timestamps correlating to 21st April 2021 around 2:45. These timestamps greatly contradict the timestamps stored in the MFT where the timestomped times date back to 2020. By reviewing the USN Journal and seeing this anomaly, an analyst should have alarm bells ringing. In my opinion, this is a more foolproof way of detecting timestomping versus looking for nanosecond precision / comparing $FN to $SI.

$Logfile Detection
It's not always feasible that the $Logfile will store the information that you're looking for – especially on busy disks. However, for the sake of showing the detection – I did this on my "not busy" VM.

What you want to look for the $Logfile are two logs pertaining to:

- Operation: CreateAttribute
- Filename: file.txt (or your filename)
- CurrentAttribute: $FILE_NAME

When performing timestomping where you're trying to overwrite the $FN attribute – there are two logs of interest in the $Logfile:

has occurred – especially when considering that the time will not match the time in the $MFT $FI and $SI attributes.

Just to show you the example from my test – this is what the two contradicting log lines look

Happy hunting!

REFERENCES

https://github.com/limbenjamin/nTimetools

http://windowsir.blogspot.com/2014/07/file-system-ops-effects-on-mft-records.html

https://github.com/jschicht/SetMace

http://forensicinsight.org/wp-content/uploads/2013/06/F-INSIGHT-NTFS-Log-TrackerEnglish.pdf

https://az4n6.blogspot.com/2014/10/timestomp-mft-shenanigans.html

https://forensicswiki.xyz/wiki/index.php?
title=Timestomp#:~:text=Timestomp%20is%20a%20utility%20co,timestamp%2Drelated%20information%20on%20files.

https://www.blackhat.com/presentations/bh-usa-05/bh-us-05-foster-liu-update.pdf

http://undocumented.ntinternals.net/index.html?
page=UserMode%2FUndocumented%20Functions%2FNT%20Objects%2FProcess%2FNtSetInformationProcess.html

$FILE_NAME timestomp timestomping timestomping $FILE_NAME

Popular posts from this blog

Forensic Analysis of AnyDesk Logs

February 10, 2021

Most threat actors during ransomware incidents utilise some type of remote access tools - one of them being AnyDesk. This is a free remote access tool that threat actors download onto hosts to access them easily and also for bidirectional file transfer. There are two locations f …

READ MORE

So you want to reverse and patch an iOS application? I got you >_< If you've missed the blogs in the series, check them out below ^_^ Part 1: How to Reverse Engineer and Patch an iOS Application for Beginners Part 2: Guide to Reversing and Exploiting iOS binaries: ARM64 RO …

READ MORE

Successful 4624 Anonymous Logons to Windows Server from External IPs?

April 30, 2020

If you see successful 4624 event logs that look a little something like this in your Event Viewer showing an ANONYMOUS LOGON, an external IP (usually from Russia, Asia, USA, Ukraine) with an authentication package of NTLM, NTLMSSP, don't be alarmed - this is not an indicat …

READ MORE