

Sherlock: BFT

■ Very Easy

Challenge Description :

In this Sherlock, you will become acquainted with MFT (Master File Table) forensics. You will be introduced to well-known tools and methodologies for analyzing MFT artifacts to identify malicious activity. During our analysis, you will utilize the MFTECmd tool to parse the provided MFT file, TimeLine Explorer to open and analyze the results from the parsed MFT, and a Hex editor to recover file contents from the MFT.

Just Answer 6 Tasks

Getting Down to the Task:

- First download the ZIP file and extract the contents of the ZIP, after that start opening the log files based on the challenges description. We are given a MFT file to analyze.
- To get what we need, I'm going to use MFTEExplorer, MFTECmd and Timeline Explorer. You can get all these tools on the Eric Zimmerman Github website [[Link](#)]
- This should be everything that we need to answer the Tasks. The MFT file will need to extract the metadata of the files contained within it.
- Using MFTECmd we convert the data contained within the MFT file to a CSV or your preferred format file, with the CSV file we can check it with the Timeline Explorer tool.

```
C:\Users\drobo\Music\MFTECmd> MFTECmd.exe -f "C:\Users\drobo\Music\MFTECmd\MFT" --csv . --csvf mft-metadata.csv
MFTECmd version 1.2.2.1

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/MFTECmd

Command line: -f C:\Users\drobo\Music\MFTECmd\MFT --csv . --csvf mft-metadata.csv

Warning: Administrator privileges not found!

File type: Mft

Processed C:\Users\drobo\Music\MFTECmd\MFT in 5.5285 seconds

C:\Users\drobo\Music\MFTECmd\MFT: FILE records found: 171,927 (Free records: 142,905) File size: 307.5MB
CSV output will be saved to .\mft-metadata.csv

C:\Users\drobo\Music\MFTECmd>
fs
Halo Theme Song ...
```

File Name	Extension	Is Directory	Has Ads	Is Ads	File Size	Created0x10	Created0x30	Last Modified0x10	Last Modified0x30
RINFO.EXE	.EXE				17488	2019-12-07 09:09:13		2019-12-07 09:09:13	
RINFO.EXE.WofCompressedData					8650	2019-12-07 09:09:13		2019-12-07 09:09:13	
RINFO.EXE	.EXE				17488	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:10:51
RINFO.EXE.WofCompressedData					8650	2019-12-07 09:09:13		2019-12-07 09:09:13	2023-07-27 12:10:51
shts.exe	.exe				14848	2019-12-07 09:09:19		2019-12-07 09:09:19	2023-07-27 12:10:51
shts.exe.WofCompressedData					5984	2019-12-07 09:09:19		2019-12-07 09:09:19	2023-07-27 12:10:51
uiUnattens.exe	.exe				106496	2019-12-07 09:08:29		2019-12-07 09:08:29	2023-07-27 12:10:51
uiUnattens.exe.WofCompressedData					55176	2019-12-07 09:08:29		2019-12-07 09:08:29	2023-07-27 12:10:51
uiUnattens.exe	.exe				83968	2019-12-07 09:09:06		2019-12-07 09:09:06	2023-07-27 12:10:51
uiUnattens.exe.WofCompressedData					47122	2019-12-07 09:09:06		2019-12-07 09:09:06	2023-07-27 12:10:51
arrnator.exe	.exe				532992	2019-12-07 09:08:21		2019-12-07 09:08:21	2023-07-27 12:10:51
arrnator.exe.WofCompressedData					236466	2019-12-07 09:08:21		2019-12-07 09:08:21	2023-07-27 12:10:51
btstat.exe	.exe				22016	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51
btstat.exe.WofCompressedData					11845	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51
btstat.exe	.exe				22016	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51
btstat.exe.WofCompressedData					11845	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51
dadmin.exe	.exe				69632	2019-12-07 09:08:46		2019-12-07 09:08:46	2023-07-27 12:10:51
dadmin.exe.WofCompressedData					24880	2019-12-07 09:08:46		2019-12-07 09:08:46	2023-07-27 12:10:51
dadmin.exe	.exe				65536	2019-12-07 09:09:11		2019-12-07 09:09:11	2023-07-27 12:10:51
dadmin.exe.WofCompressedData					22573	2019-12-07 09:09:11		2019-12-07 09:09:11	2023-07-27 12:10:51
et.exe	.exe				59984	2019-12-07 09:09:13		2019-12-07 09:09:13	2023-07-27 12:10:51
et.exe.WofCompressedData					34110	2019-12-07 09:09:13		2019-12-07 09:09:13	2023-07-27 12:10:51
et.exe	.exe				59984	2019-12-07 09:09:13		2019-12-07 09:09:13	2023-07-27 12:10:51
et.exe.WofCompressedData					34110	2019-12-07 09:09:13		2019-12-07 09:09:13	2023-07-27 12:10:51
etl.exe	.exe				183808	2019-12-07 09:09:13		2019-12-07 09:09:13	2023-07-27 12:10:51
etl.exe.WofCompressedData					103729	2019-12-07 09:09:13		2019-12-07 09:09:13	2023-07-27 12:10:51
etl.exe	.exe				27136	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51
etl.exe.WofCompressedData					14485	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51
etl.exe	.exe				37376	2019-12-07 09:08:33		2019-12-07 09:08:33	2023-07-27 12:10:51
etl.exe.WofCompressedData					19731	2019-12-07 09:08:33		2019-12-07 09:08:33	2023-07-27 12:10:51
etl.exe	.exe				37376	2019-12-07 09:08:33		2019-12-07 09:08:33	2023-07-27 12:10:51
etl.exe.WofCompressedData					19731	2019-12-07 09:08:33		2019-12-07 09:08:33	2023-07-27 12:10:51
etcfghotiffyObjectHos.exe	.exe				75776	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51
etcfghotiffyObjectHos.exe.WofCompressedData					30170	2019-12-07 09:08:44		2019-12-07 09:08:44	2023-07-27 12:10:51

- Opening it up with Timeline Explorer, we now need to filter it and get the answers to the Tasks we need to answer. This is pretty simple: the first task is to just search on the Day it said it occurred and the file type he said he downloaded.

ast Record	Change0x30	Last Access0x10	Last Access0x30	Zone Id	Contents	Parse Target	Reference Count	SI<FN	u Sec Zeros	Copied
024-02-13 16:43:46		2024-02-13 16:44:55	2024-02-13 16:43:46	[ZoneTransfer] ZoneId=3	RefererUrl=https://justbeamit.com/ HostUrl=https://e.com .justbeamit.com:8443/download?token=yyma5		1			
024-02-13 16:38:39		2024-02-13 16:38:41	2024-02-13 16:38:39	[ZoneTransfer] ZoneId=3	RefererUrl=C:\Use rs\simon.stark\Down loads\Stage-20240 213T093324Z-001\St age\invoice\invoic es.zip		1			
024-02-13 16:35:31		2024-02-13 16:38:39	2024-02-13 16:35:31	[ZoneTransfer] ZoneId=3	RefererUrl=C:\Use rs\simon.stark\Down loads\Stage-20240 213T093324Z-001\St age\invoice.zip		1			

- The second task is to determine the origin of the file. This will be straightforward using the Timeline Explorer. The CSV file contains various headers or titles, which the Timeline Explorer uses to organize the data like an Excel sheet.
- The title "Zone.Id" is crucial for identifying the file's origin because, when a file is downloaded on Windows, it is tagged with a label called Zone.Identifier. This label indicates where the file came from, this only applies to online downloads.
- The rest of the tasks are the same: To find a file and to analyze the \$Created0x30 timestamp to find when the file was created.

- For the last two tasks you will need to find the hex offset of an MFT record. Then find the IP:PORT of a malicious C2 that infected the machine.
- The hex offset requires some math. An entry is 1024 bytes. The file size is 23,436 bytes. Knowing both of these, we can multiply them to find the total size in bytes, and then converting to hexadecimal for the offset:

$23,436 \text{ entries} * 1024 \text{ bytes/entry} = 23,984,640 \text{ bytes} = 0x16E3000 \text{ Hex}$

- To get the C2 IP:PORT you will need to open the MFT file with a Hex Editor to go to that Hexadecimal offset. I used HxD for this to work.

The screenshot shows the HxD hex editor interface. On the left, a list of MFT records is visible, with addresses 016E2FEB through 016E32DF. The main area displays hex data in columns, with corresponding ASCII text on the right. A 'Go to' dialog box is open in the center, with the 'Offset' field set to '16E3000'. The 'Offset relative to' section has 'begin' selected. The 'OK' button is highlighted.

- Searching up the Offset in HxD should provide us the IP:PORT to the C2 that Infected the machine.

Task 1

Simon Stark was targeted by attackers on February 13. He downloaded a ZIP file from a link received in an email. What was the name of the ZIP file he downloaded from the link?

Stage-20240213T093324Z-001.zip ✓

Task 2

Examine the Zone Identifier contents for the initially downloaded ZIP file. This field reveals the HostUrl from where the file was downloaded, serving as a valuable Indicator of Compromise (IOC) in our investigation/analysis. What is the full Host URL from where this ZIP file was downloaded?

https://storage.googleapis.com/drive-bulk-export-anonymous/20240213T09 ✓

Task 3 Hint

What is the full path and name of the malicious file that executed malicious code and connected to a C2 server?

C:\Users\simon.stark\Downloads\Stage-20240213T093324Z-001\Stage\im ✓

Task 4

Analyze the \$Created0x30 timestamp for the previously identified file. When was this file created on disk?

2024-02-13 16:38:39 ✓

Task 5 Hint

Finding the hex offset of an MFT record is beneficial in many investigative scenarios. Find the hex offset of the stager file from Question 3.

16E3000 ✓

Task 6 Hint

Each MFT record is 1024 bytes in size. If a file on disk has smaller size than 1024 bytes, they can be stored directly on MFT File itself. These are called MFT Resident files. During Windows File system investigation, it's crucial to look for any malicious/suspicious files that may be resident in MFT. This way we can find contents of malicious files/scripts. Find the contents of the malicious stager identified in Question 3 and answer with the C2 IP and port.

43.204.110.203:6666 ✓