Sherlock: BFT

■ Very Easy

Challenge Description :

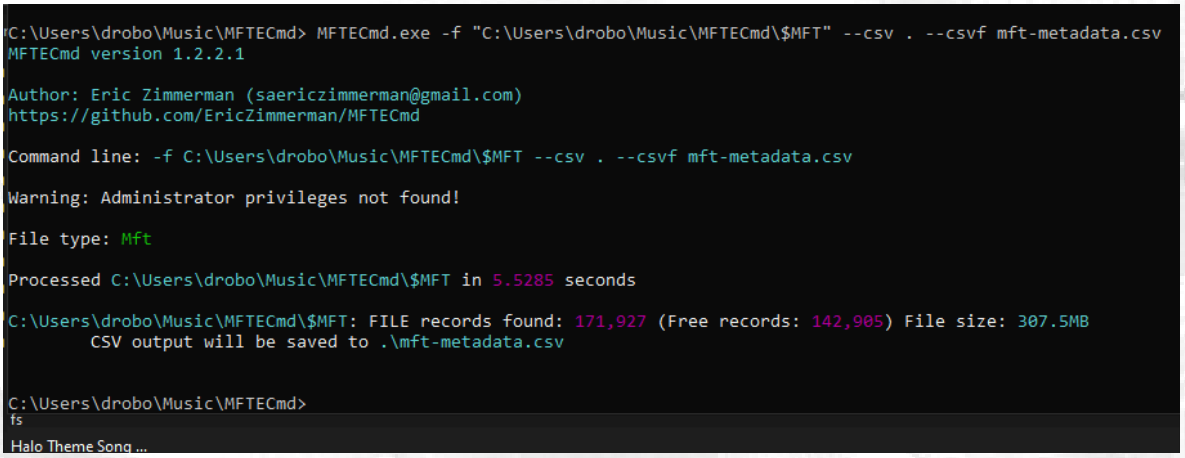
In this Sherlock, you will become acquainted with MFT (Master File Table) forensics. You will be introduced to well-known tools and methodologies for analyzing MFT artifacts to identify malicious activity. During our analysis, you will utilize the MFTECmd tool to parse the provided MFT file, TimeLine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

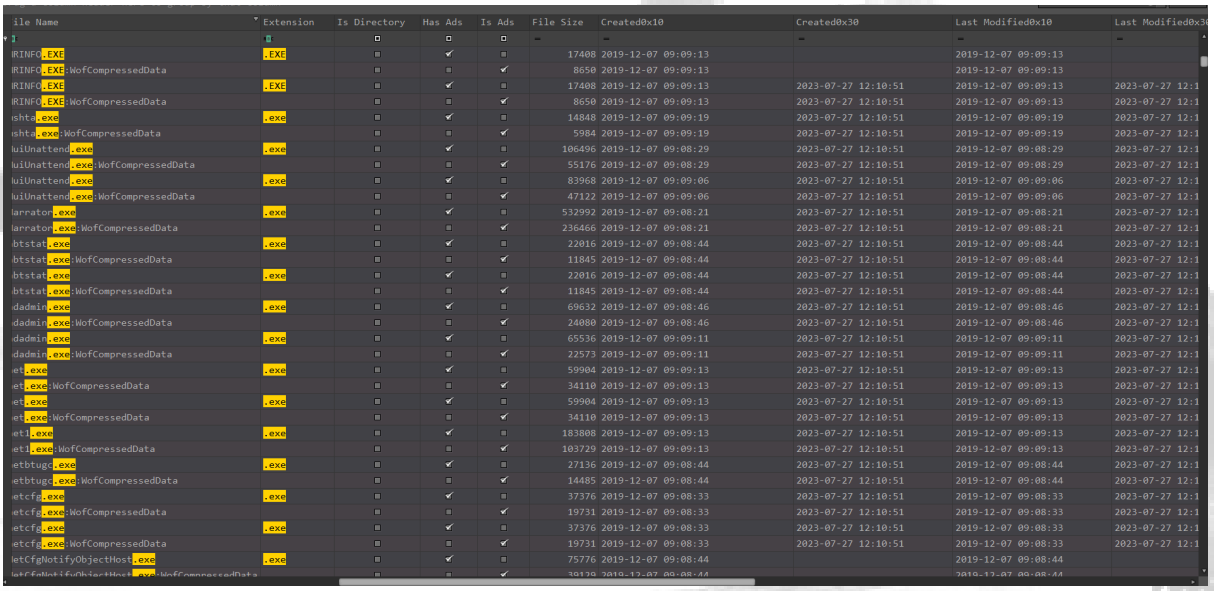
Just Answer 6 Tasks

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.

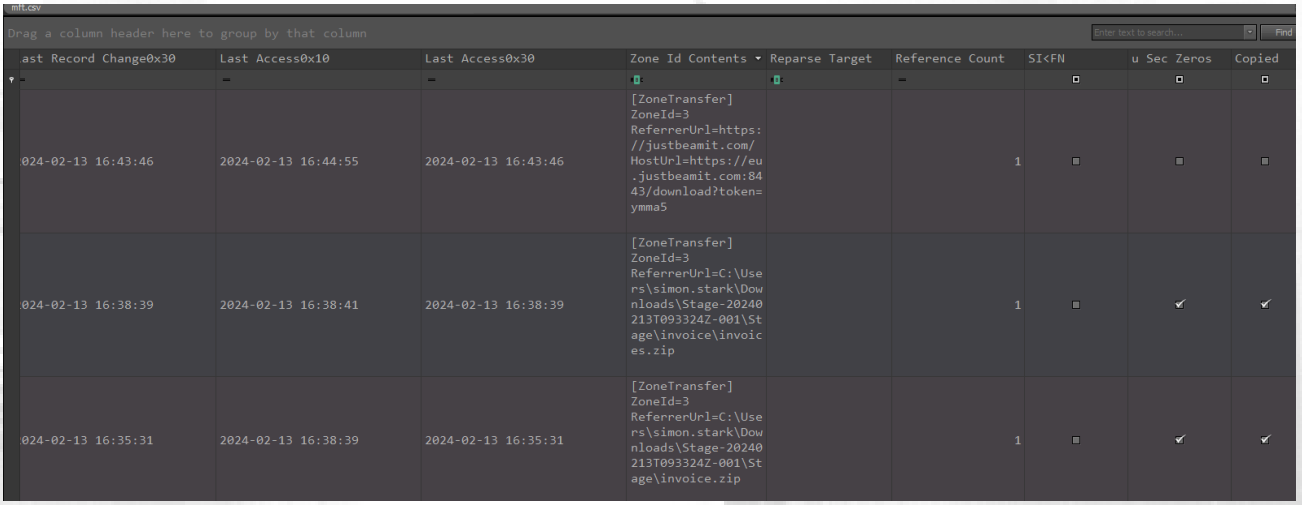
- To get what we need, I'm going to use MFTExplorer, MFTECmd and Timeline Explorer. You can get all these tools on the Eric Zimmerman Github website [Link]
- This should be everything that we need to answer the Tasks. The MFT file will need to extract the metadata of the files contained within it.
- 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.





ile Name	Extension	Is Directory	Has Ads	Is Ads	File Size	Created0x10	Created0x30	Last Modified0x10	Last Modified0x3
IRINFO.EXE	.EXE	☐	☑	☐	17408	2019-12-07 09:09:13		2019-12-07 09:09:13	
IRINFO.EXE:WofCompressedData		☐	☐	☑	8650	2019-12-07 09:09:13		2019-12-07 09:09:13	
IRINFO.EXE	.EXE	☐	☑	☐	17408	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
IRINFO.EXE:WofCompressedData		☐	☐	☑	8650	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
ishta.exe	.exe	☐	☑	☐	14848	2019-12-07 09:09:19	2023-07-27 12:10:51	2019-12-07 09:09:19	2023-07-27 12:1
ishta.exe:WofCompressedData		☐	☐	☑	5984	2019-12-07 09:09:19	2023-07-27 12:10:51	2019-12-07 09:09:19	2023-07-27 12:1
luiUnattend.exe	.exe	☐	☑	☐	106496	2019-12-07 09:08:29	2023-07-27 12:10:51	2019-12-07 09:08:29	2023-07-27 12:1
luiUnattend.exe:WofCompressedData		☐	☐	☑	55176	2019-12-07 09:08:29	2023-07-27 12:10:51	2019-12-07 09:08:29	2023-07-27 12:1
luiUnattend.exe	.exe	☐	☑	☐	83968	2019-12-07 09:09:06	2023-07-27 12:10:51	2019-12-07 09:09:06	2023-07-27 12:1
luiUnattend.exe:WofCompressedData		☐	☐	☑	47122	2019-12-07 09:09:06	2023-07-27 12:10:51	2019-12-07 09:09:06	2023-07-27 12:1
larrator.exe	.exe	☐	☑	☐	532992	2019-12-07 09:08:21	2023-07-27 12:10:51	2019-12-07 09:08:21	2023-07-27 12:1
larrator.exe:WofCompressedData		☐	☐	☑	236466	2019-12-07 09:08:21	2023-07-27 12:10:51	2019-12-07 09:08:21	2023-07-27 12:1
btstat.exe	.exe	☐	☑	☐	22016	2019-12-07 09:08:44	2023-07-27 12:10:51	2019-12-07 09:08:44	2023-07-27 12:1
btstat.exe:WofCompressedData		☐	☐	☑	11845	2019-12-07 09:08:44	2023-07-27 12:10:51	2019-12-07 09:08:44	2023-07-27 12:1
btstat.exe	.exe	☐	☑	☐	22016	2019-12-07 09:08:44	2023-07-27 12:10:51	2019-12-07 09:08:44	2023-07-27 12:1
btstat.exe:WofCompressedData		☐	☐	☑	11845	2019-12-07 09:08:44	2023-07-27 12:10:51	2019-12-07 09:08:44	2023-07-27 12:1
dadmin.exe	.exe	☐	☑	☐	69632	2019-12-07 09:08:46	2023-07-27 12:10:51	2019-12-07 09:08:46	2023-07-27 12:1
dadmin.exe:WofCompressedData		☐	☐	☑	24080	2019-12-07 09:08:46	2023-07-27 12:10:51	2019-12-07 09:08:46	2023-07-27 12:1
dadmin.exe	.exe	☐	☑	☐	65536	2019-12-07 09:09:11	2023-07-27 12:10:51	2019-12-07 09:09:11	2023-07-27 12:1
dadmin.exe:WofCompressedData		☐	☐	☑	22573	2019-12-07 09:09:11	2023-07-27 12:10:51	2019-12-07 09:09:11	2023-07-27 12:1
et.exe	.exe	☐	☑	☐	59904	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
et.exe:WofCompressedData		☐	☐	☑	34110	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
et.exe	.exe	☐	☑	☐	59904	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
et.exe:WofCompressedData		☐	☐	☑	34110	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
et1.exe	.exe	☐	☑	☐	183808	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
et1.exe:WofCompressedData		☐	☐	☑	103729	2019-12-07 09:09:13	2023-07-27 12:10:51	2019-12-07 09:09:13	2023-07-27 12:1
etbtugc.exe	.exe	☐	☑	☐	27136	2019-12-07 09:08:44	2023-07-27 12:10:51	2019-12-07 09:08:44	2023-07-27 12:1
etbtugc.exe:WofCompressedData		☐	☐	☑	14485	2019-12-07 09:08:44	2023-07-27 12:10:51	2019-12-07 09:08:44	2023-07-27 12:1
etcfg.exe	.exe	☐	☑	☐	37376	2019-12-07 09:08:33	2023-07-27 12:10:51	2019-12-07 09:08:33	2023-07-27 12:1
etcfg.exe:WofCompressedData		☐	☐	☑	19731	2019-12-07 09:08:33	2023-07-27 12:10:51	2019-12-07 09:08:33	2023-07-27 12:1
etcfg.exe	.exe	☐	☑	☐	37376	2019-12-07 09:08:33	2023-07-27 12:10:51	2019-12-07 09:08:33	2023-07-27 12:1
etcfg.exe:WofCompressedData		☐	☐	☑	19731	2019-12-07 09:08:33	2023-07-27 12:10:51	2019-12-07 09:08:33	2023-07-27 12:1
etCfgNotifyObjectHost.exe	.exe	☐	☑	☐	75776	2019-12-07 09:08:44		2019-12-07 09:08:44	
etCfgNotifyObjectHost.exe:WofCompressedData		☐	☐	☑	39128	2019-12-07 09:08:44		2019-12-07 09:08:44	

- 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.



mft.csv

Drag a column header here to group by that column

Enter text to search... Find

ast Record Change0x30	Last Access0x10	Last Access0x30	Zone Id Contents	Reparse 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 ReferrerUrl=https://justbeamit.com/ HostUrl=https://eu.justbeamit.com:8443/download?token=ymma5		1	☐	☐	☐
024-02-13 16:38:39	2024-02-13 16:38:41	2024-02-13 16:38:39	[ZoneTransfer] ZoneId=3 ReferrerUrl=C:\Users\simon.stark\Downloads\Stage-20240213T093324Z-001\Stage\invoice\invoices.zip		1	☐	☑	☑
024-02-13 16:35:31	2024-02-13 16:38:39	2024-02-13 16:35:31	[ZoneTransfer] ZoneId=3 ReferrerUrl=C:\Users\simon.stark\Downloads\Stage-20240213T093324Z-001\Stage\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.

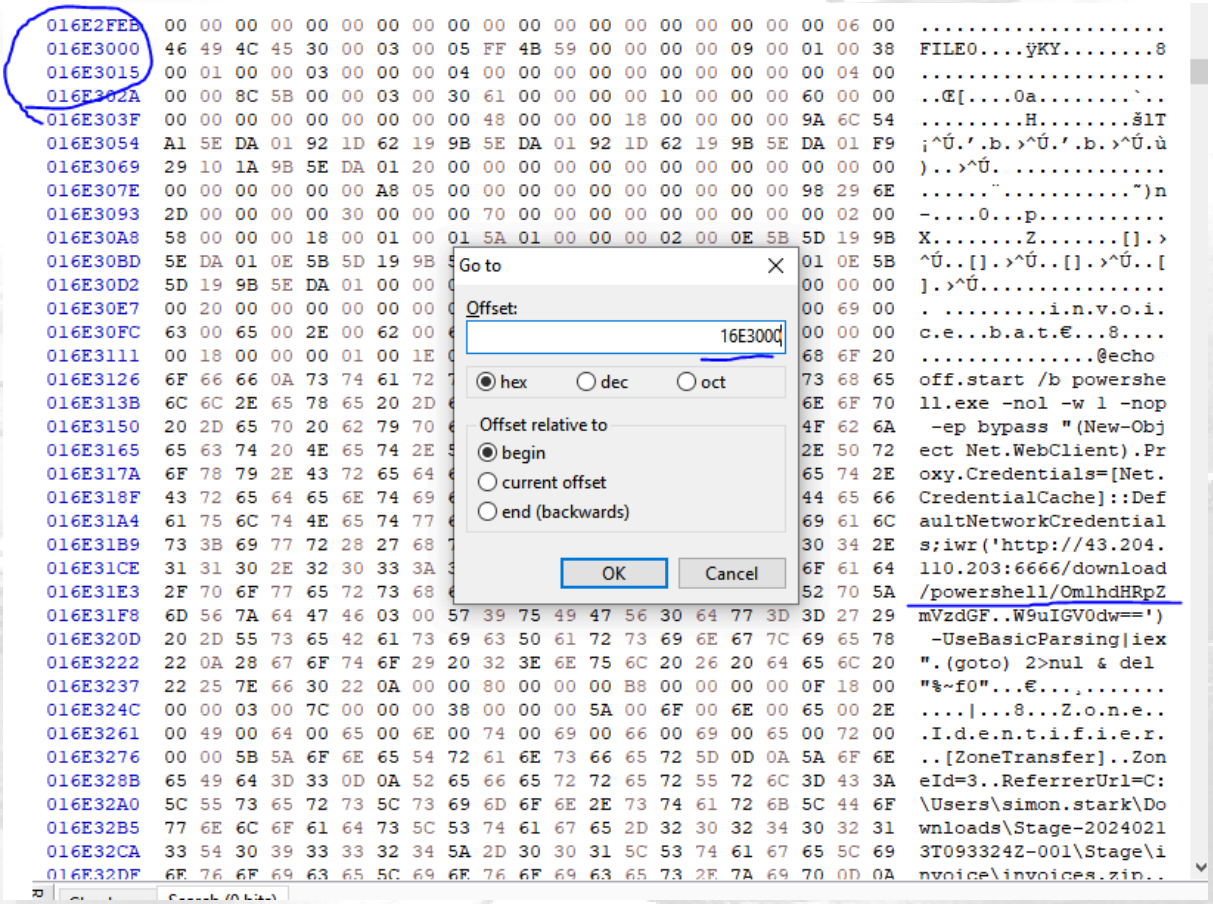
- The rest of the tasks are the same: To find a file and to analyze the $Created0x30 timestamp to find when the file was created.

- 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.

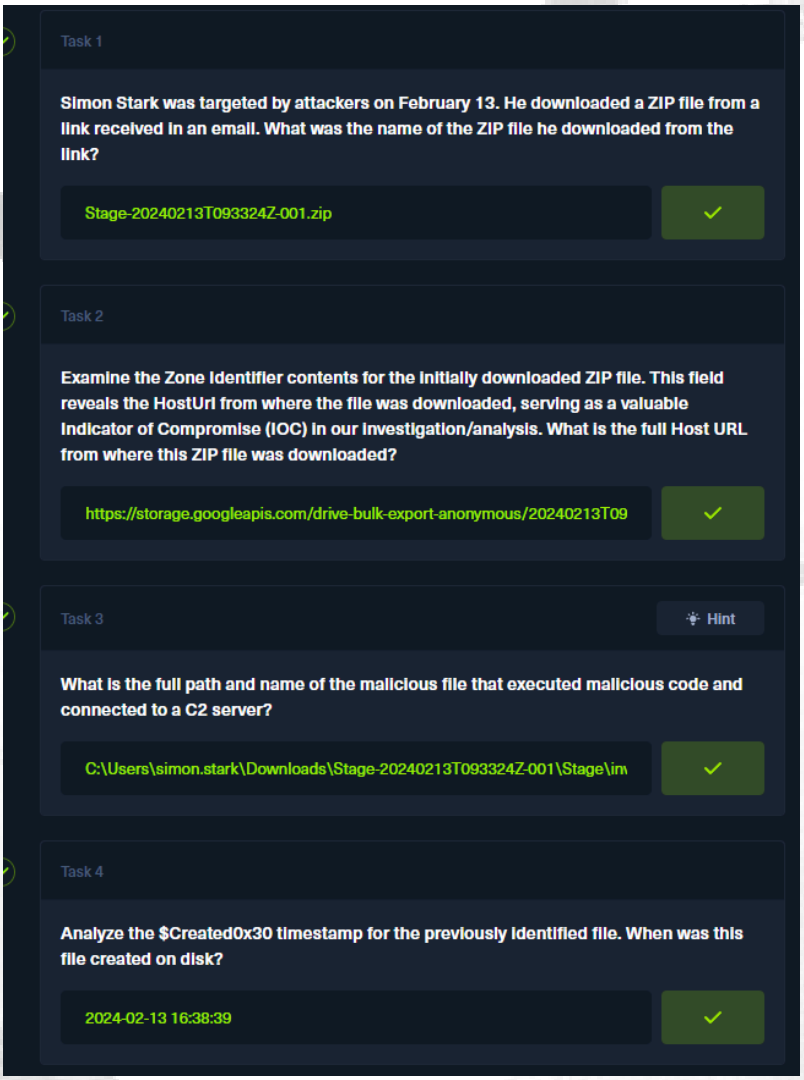
- The hex offset requires some math. An entry is 1024 bytes. The file size is 23,436 bytes. Knowing both of these, we can multiply them to find the total size in bytes, and then converting to hexadecimal for the offset:

23,436 entries * 1024 bytes/entry = 23,984,640 bytes = 0x16E3000 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.



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



Sherlock BFT - Birdo