

# 3주차 스터디

## 이전 수업 실습 + 파일 시스템 분석 + KEEPER CTF IR-1 문제 풀어보기

1~2주차는 온라인으로 진행하였으며, 3주차부터 대면으로 진행  
원활한 실습 진행을 위해 1~2주차에서 사용된 슬라이드를 일부 추가하였습니다.

# KEEPER CTF IR-1 문제 설명 및 다운로드 링크

랜섬웨어에 걸린 것 같다.

최초로 감염된 파일명과 감염 시간, 어떤 파일에 의해 감염되었는지 입력하라.

파일명은 소문자로 입력, 띄어쓰기는 언더바(\_) 처리, 타임스탬프는 한국 시간인 UTC+9를 따르며, ISO 8601 표준에 의해 날짜와 날짜 사이에 T 문자를 입력한다.

ex: KEEPER{asdf.asd\_2024-12-23T12:34:56\_example.exe}

Download Link: <https://drive.google.com/file/d/1KhkiZXagtpBXRQ63et2ZCyDtpvAlko87>

**용량이 크기 때문에 스터디 시작 전에 미리 받아두기**

# E01이 뭘까?

Google Drive에서 파일에 바이러스가 있는지 검사할 수 없습니다

240103.E01(6.3G) 파일이 너무 커서 바이러스 검사를 할 수 없습니다. 그래도 파일을 다운로드하시겠습니까?

무시하고 다운로드

- 문제 파일을 보면 E01이라는 확장자와 용량(6.3G)이 큰 것을 확인할 수 있다.

## Introduction

Developed by ASR Data, the Expert Witness file format (aka E01 format aka EnCase file format) is an industry standard format for storing “forensic” images. The format allows a user to access arbitrary offsets in the uncompressed data without requiring decompression of the entire data stream. The specification does **NOT** provide for quantifiable assurance of integrity, it is up to the implementation to provide meaningful authentication for **any** data contained in an “evidence file”.

- E01은 Guidance software<sup>1</sup>에서 개발한 압축 포맷이며, 하드디스크 백업본을 파일의 형태로 저장할 수 있다. 대부분의 포렌식 이미징 도구에서 E01 파일 포맷을 지원한다.<sup>234</sup>

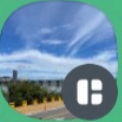
<sup>1</sup> [https://en.wikipedia.org/wiki/Guidance\\_Software](https://en.wikipedia.org/wiki/Guidance_Software)

<sup>2</sup> [https://forensics.wiki/encase\\_image\\_file\\_format/](https://forensics.wiki/encase_image_file_format/)

<sup>3</sup> <https://blog.naver.com/happymaru11/222102005996>

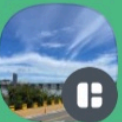
<sup>4</sup> [http://www.asrdata.com/?page\\_id=1566](http://www.asrdata.com/?page_id=1566)

# E01이 뭘까?

 Forensicator


대부분 이미징할때도 E01쓰고, 보관할 때  
도 E01 그대로 보관합니다.

49  
20:36

 Forensicator

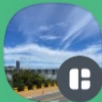
RAW는 E01으로 할 수 없는 경우에 제한  
적으로 사용되고, 현장에서 바로 E01으로  
변환하거나 보관할 때 E01 혹은 압축해서  
보관합니다

49  
20:37

 Forensicator

E01(EWF 포맷)은 EnCase가 시장 점유  
율이 굉장히 높던 15년 전에 사용된 포맷으  
로 높은 시장점유율로 포렌식 업계에서 사  
실상 표준으로 쓰여왔습니다.

49  
20:41

 Forensicator

10년 전부터는 EnCase의 시장 지배력이  
사라져 지금은 새로운 포맷인 Ex01도 표준  
으로 안쓰이고 L01도 EnCase 외에는  
다른 시장에서 거의 지원안합니다.  
최근 논리이미징은 오픈소스 포맷인  
AFF4-L이나 압축 형식이 많이 사용됩니  
다

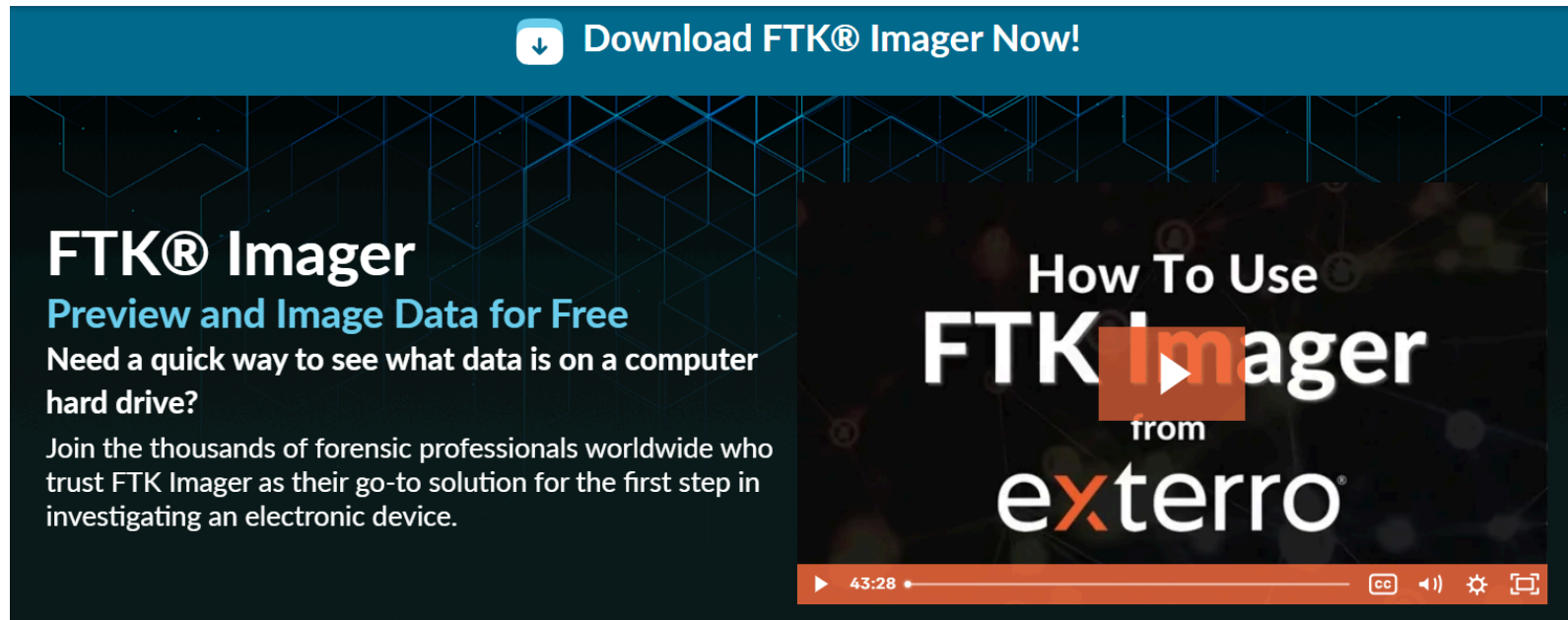
49  
20:43

✓ 1

출처: 디지털 포렌식 정보공유 오픈채팅방

# FTK Imager 사용하기

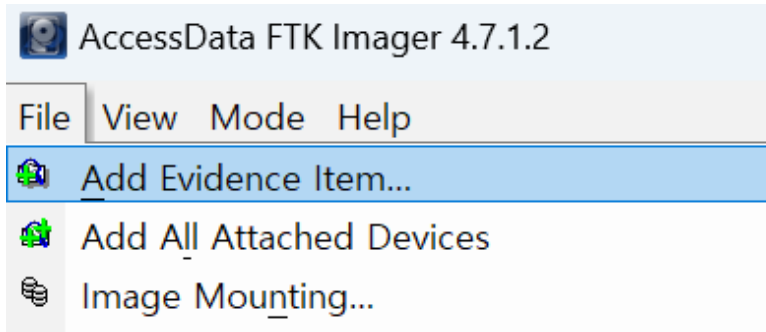
- E01 파일을 열기 위해서 해당 포맷을 해석해주는 도구를 사용해야 한다.
- 다양한 이미징 도구 중에서 무료인 FTK Imager를 많이 사용한다.<sup>1</sup>
  - 나중에 [Arsenal Image Mounter](#)도 사용해보는 것을 추천한다.
- Download Link: <https://www.exterro.com/ftk-imager>



<sup>1</sup> [https://en.wikipedia.org/wiki/Digital\\_forensic\\_process#Acquisition](https://en.wikipedia.org/wiki/Digital_forensic_process#Acquisition)

# FTK Imager 사용하기

- File → Add Evidence Item → Image File → E01 파일 열기



Add Evidence Item 클릭

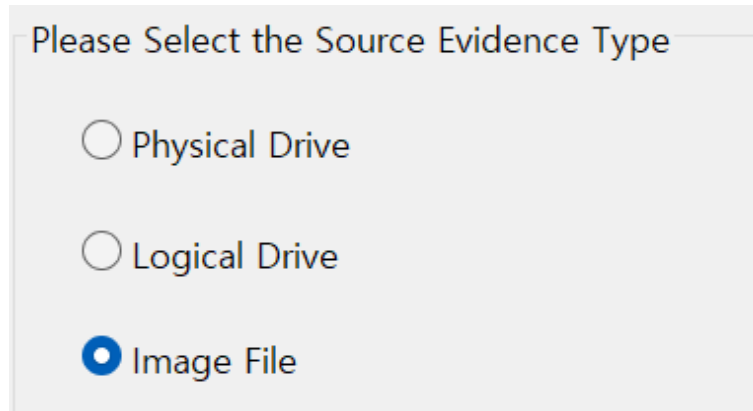
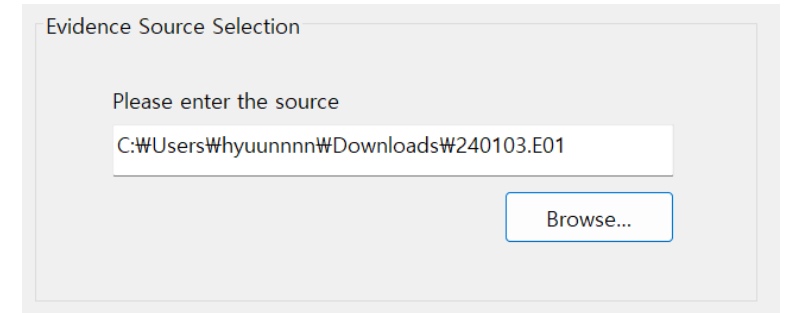


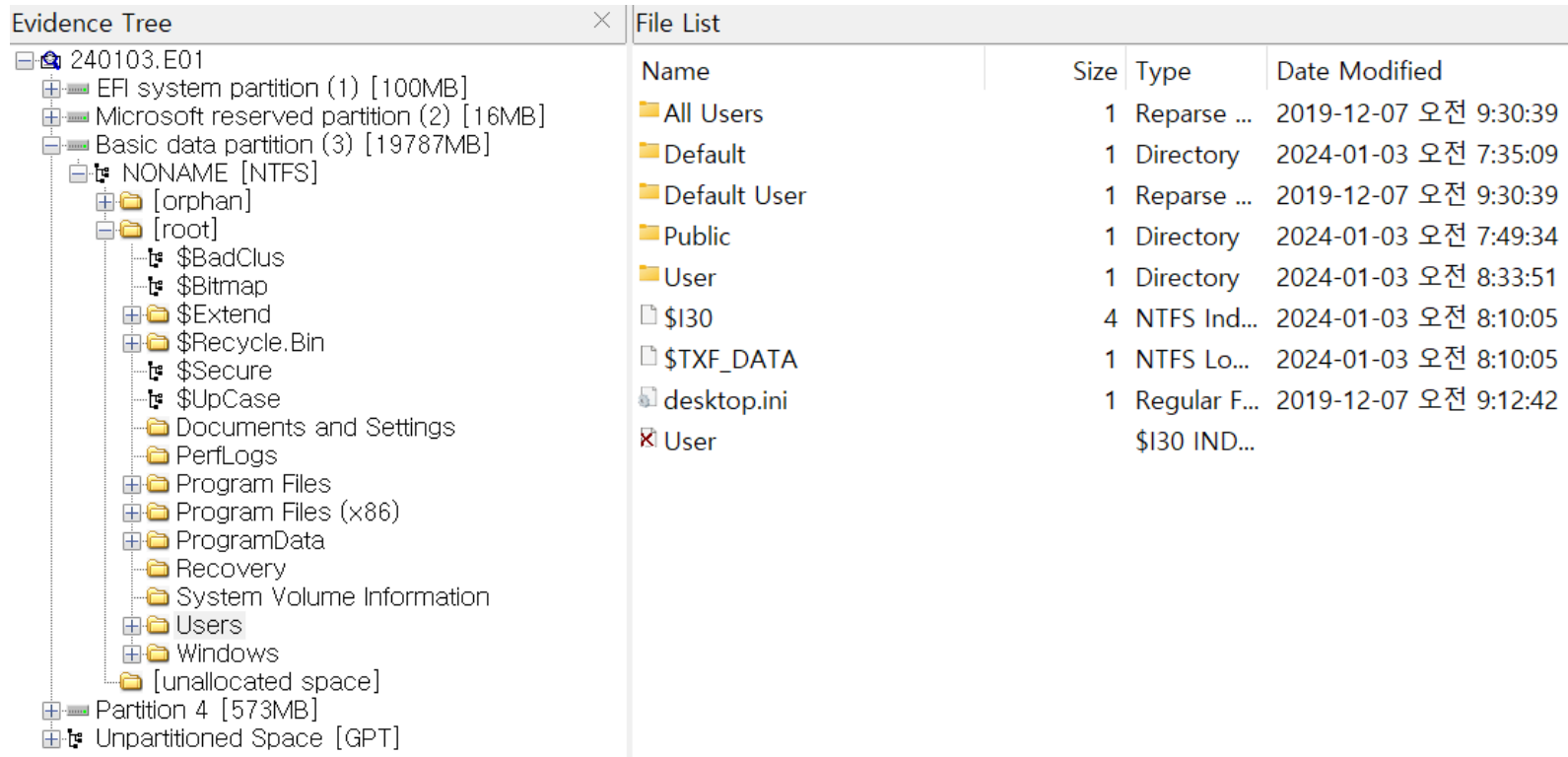
Image File 클릭



E01 파일 열기

# FTK Imager 사용하기

- 윈도우를 포맷할 때 사용되는 파티션 외에도 시스템 예약 파티션과 같은 추가 파티션이 생성된다.
- 그 중에서 C 드라이브의 파티션을 확인해보자. (용량이 가장 큰 파티션을 누르면 된다.)
- 파티션에 존재하는 파일들을 추출할 수 있다.



The screenshot displays the FTK Imager interface with two main panes: 'Evidence Tree' on the left and 'File List' on the right.

**Evidence Tree:** Shows a hierarchical view of the disk image. The root is '240103.E01'. It contains several partitions: 'EFI system partition (1) [100MB]', 'Microsoft reserved partition (2) [16MB]', 'Basic data partition (3) [19787MB]', 'Partition 4 [573MB]', and 'Unpartitioned Space [GPT]'. The 'Basic data partition (3)' is expanded, showing a file system structure with folders like '[orphan]', '[root]', '\$BadClus', '\$Bitmap', '\$Extend', '\$Recycle.Bin', '\$Secure', '\$UpCase', 'Documents and Settings', 'PerfLogs', 'Program Files', 'Program Files (x86)', 'ProgramData', 'Recovery', 'System Volume Information', 'Users', 'Windows', and '[unallocated space]'.

**File List:** A table showing the contents of the selected partition. The columns are 'Name', 'Size', 'Type', and 'Date Modified'.

Name	Size	Type	Date Modified
All Users	1	Reparse ...	2019-12-07 오전 9:30:39
Default	1	Directory	2024-01-03 오전 7:35:09
Default User	1	Reparse ...	2019-12-07 오전 9:30:39
Public	1	Directory	2024-01-03 오전 7:49:34
User	1	Directory	2024-01-03 오전 8:33:51
\$I30	4	NTFS Ind...	2024-01-03 오전 8:10:05
\$TXF_DATA	1	NTFS Lo...	2024-01-03 오전 8:10:05
desktop.ini	1	Regular F...	2019-12-07 오전 9:12:42
User		\$I30 IND...	

# 이전 수업 실습

- Registry
  - Registry Explorer + RECcmd
  - REGA
- Event Log
  - Glossy
  - EvtxECmd
- CSV 분석
  - Excel 프로그램
  - Timeline Explorer



# 레지스트리 구조

- 해당 파일들이 모여서 레지스트리 구조를 구성한다.

레지스트리 경로	파일 경로
HKLM\SYSTEM	%WINDIR%\SYSTEM32\Config\SYSTEM
HKLM\SAM	%WINDIR%\SYSTEM32\Config\SAM
HKLM\SECURITY	%WINDIR%\SYSTEM32\Config\SECURITY
HKLM\SOFTWARE	%WINDIR%\SYSTEM32\Config\SOFTWARE
HKEY_USERS\{User SID}	%UserProfile%\NTUSER.DAT
HKEY_USERS\{User SID}_Classes	%UserProfile%\AppData\Local \Microsoft\Windows\UsrClass.dat

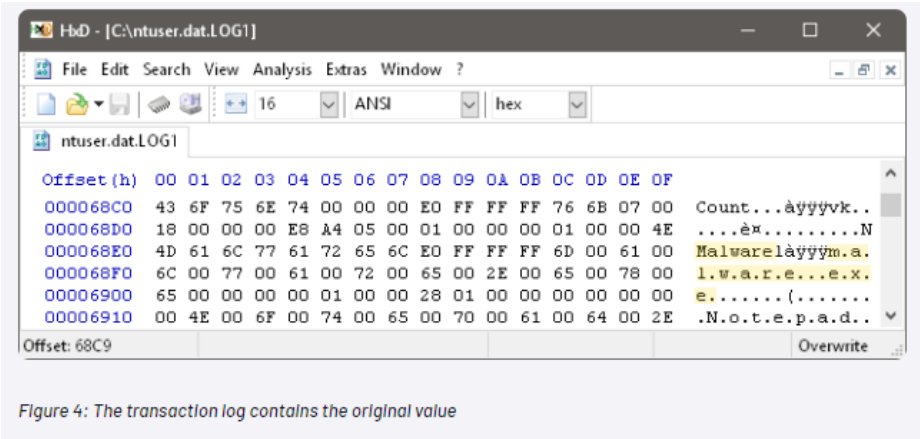
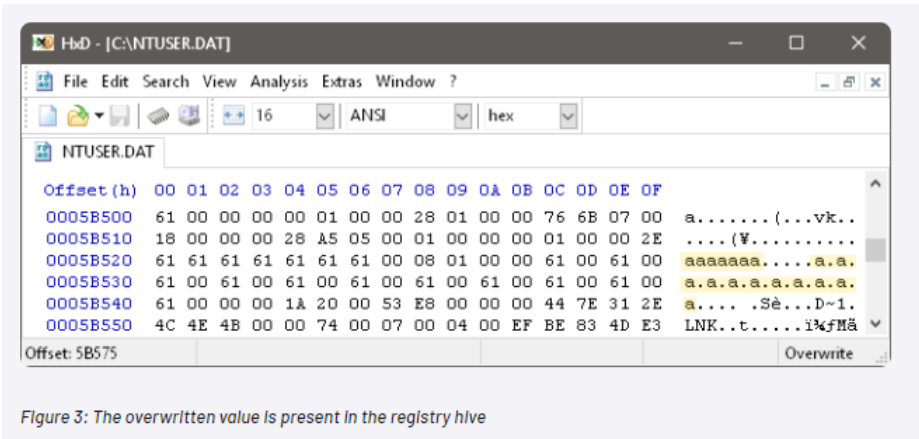
- NTUSER.DAT , UsrClass.dat 파일은 각 유저마다 별도의 파일로 존재

# LOG1, LOG2 파일은 뭘까?

NTUSER.DAT	1,024	Regular File	2024-01-03 오전 8:16:...
NTUSER.DAT.FileSlack	40	File Slack	
ntuser.dat.LOG1	328	Regular File	2024-01-03 오전 7:48:...
ntuser.dat.LOG2	564	Regular File	2024-01-03 오전 7:48:...

SYSTEM.LOG2	0	Regular File	2019-12-07 오전 9:03:...
SYSTEM.LOG1	1,328	Regular File	2019-12-07 오전 9:03:...
SYSTEM	11,520	Regular File	2024-01-03 오전 8:16:...
SOFTWARE.LOG2	1,856	Regular File	2019-12-07 오전 9:03:...
SOFTWARE.LOG1	4,880	Regular File	2019-12-07 오전 9:03:...
SOFTWARE	68,608	Regular File	2024-01-03 오전 8:16:...
SECURITY.LOG2	67	Regular File	2019-12-07 오전 9:03:...
SECURITY.LOG1	24	Regular File	2019-12-07 오전 9:03:...
SECURITY	32	Regular File	2024-01-03 오전 8:16:...
SAM.LOG2	48	Regular File	2019-12-07 오전 9:03:...
SAM.LOG1	64	Regular File	2019-12-07 오전 9:03:...
SAM	64	Regular File	2024-01-03 오전 8:16:...

- 레지스트리 하이브에 저장하기 전에 보류 중인 트랜잭션 로그이다.



출처: Digging Up the Past: Windows Registry Forensics Revisited

# LOG1, LOG2 파일은 뭘까?

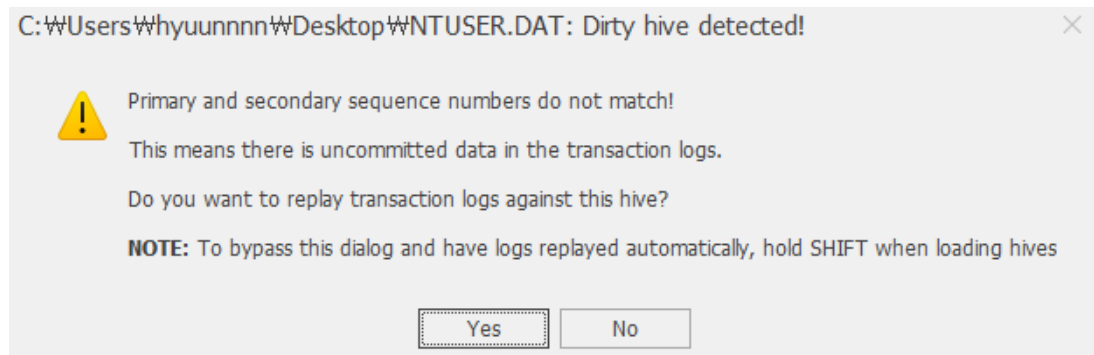
- 하이브 파일( SOFTWARE , SYSTEM , NTUSER.DAT , ...)과 LOG1 , LOG2 파일을 모두 추출한 후 Dirty 상태에서 Clean한 파일로 생성하는 과정이 필요하다. (하이브 파일과 트랜잭션 파일들을 합치는 작업)

## Types of files

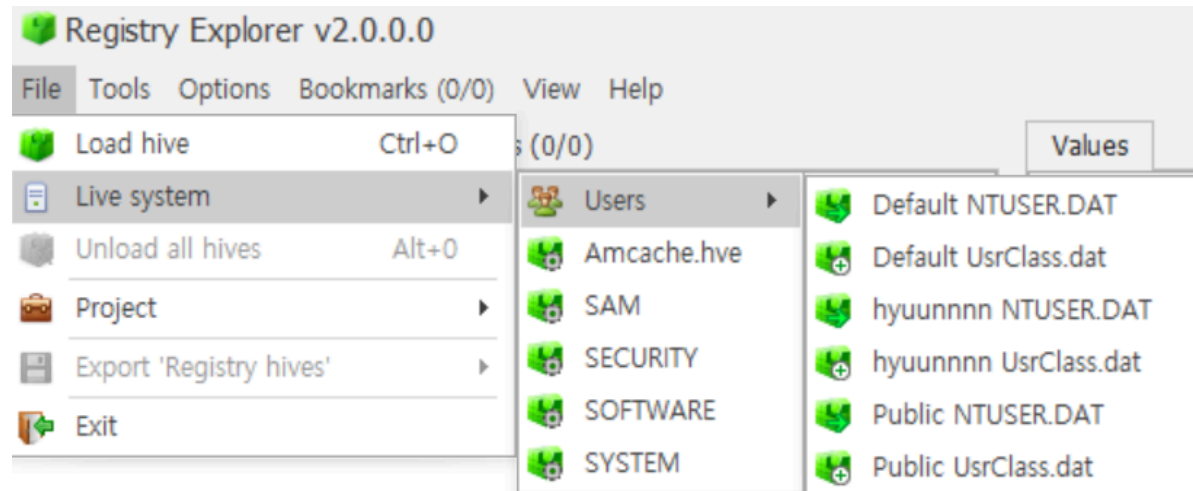
Stable registry hives consist of primary files, transaction log files, and backup copies of primary files. Primary files and their backup copies share the same format to hold actual data making up a Windows registry, transaction log files are used to perform fault-tolerant writes to primary files. Before writing modified (dirty) data to a primary file, a hive writer will store this data in a transaction log file. If an error (like a system crash) occurs when writing to a transaction log file, a primary file will remain consistent; if an error occurs when writing to a primary file, a transaction log file will contain enough data to recover a primary file and bring it back to the consistent state.

출처: [Windows registry file format specification](#)

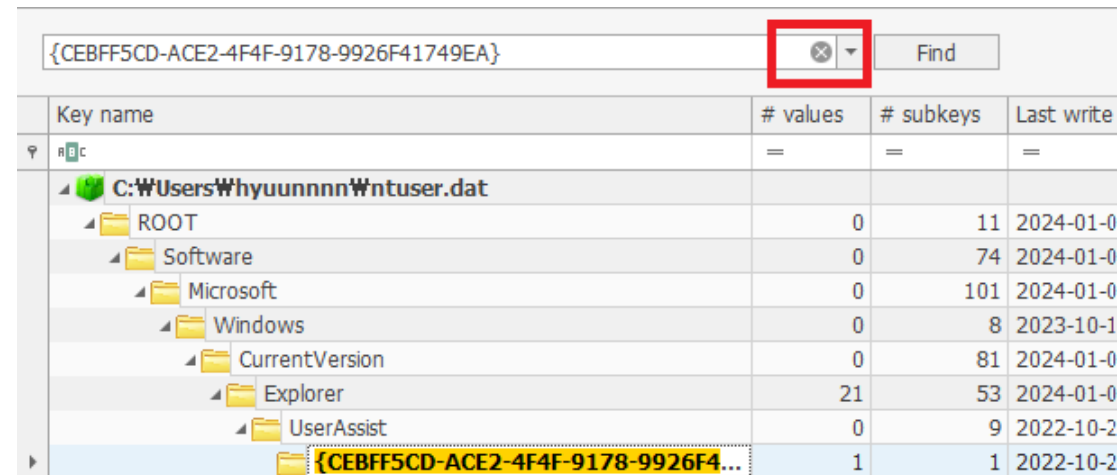
- Registry Explorer에서도 트랜잭션 로그들을 합쳐서 Clean한 파일을 만들 것인지 알려준다.



# Registry Explorer + RECmd



- ex:) UserAssist 경로 검색 → 경로 클릭 → X 버튼 클릭 → 이후에 존재하는 경로 탐색



# Registry Explorer + RECmd

- 검색한 경로까지만 뜨기 때문에 경로 클릭 후 X 버튼 클릭
- 해당 경로 이후에 존재하는 **Count** 를 클릭하여 확인 가능

The screenshot shows the Registry Explorer v2.0.0.0 interface. The left pane displays the registry tree with the path `HKEY_CURRENT_USER\Software\Classes\CLSID\{...}\Count` selected. The right pane shows the values for this path, including `UEME_CTLCUACount:ctor`, `UEME_CTLSESSION`, and various application paths like `Microsoft.Windows.Calculator_8wekyb3d8bbwe!App`.

Key name	# values	# subkeys	Last write
<code>HKEY_CURRENT_USER\Software\Classes\CLSID\{...}</code>	=	=	=
<code>UEME_CTLCUACount:ctor</code>	0	1	2022-
<code>UEME_CTLSESSION</code>	26	0	2024-
<code>Microsoft.Windows.Calculator_8wekyb3d8bbwe!App</code>	20	0	2022-
<code>Microsoft.Paint_8wekyb3d8bbwe!App</code>	0	9	2022-
<code>Microsoft.WindowsNotepad_8wekyb3d8bbwe!App</code>	1	1	2022-
<code>Microsoft.Windows.Client.CBS_cw5n1h2txyewy!CortanaUI</code>	1	1	2022-
<code>MSEdge</code>	1	1	2022-
<code>windows.immersivecontrolpanel_cw5n1h2txyewy!microsoft.windows.immersivecontrolpanel</code>	354	0	2024-
<code>{System}Wcmd.exe</code>	1	1	2022-
<code>Microsoft.Windows.Explorer</code>	1	1	2022-
<code>{ProgramFilesX64}WBandizipWBandizip.exe</code>	1	1	2022-
<code>Microsoft.VisualStudioCode</code>	3	1	2024-
<code>Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy!App</code>	0	19	2022-
<code>VirtualDesktops</code>	7	0	2024-
<code>VisualEffects</code>	18	1	2023-
<code>Wallpapers</code>	0	0	2022-
<code>WordWheelQuery</code>	0	0	2022-
<code>Ext</code>	0	0	2022-

# Registry Explorer + RECmd

- Available bookmarks 버튼을 누르면 현재 적용된 북마크 사용 가능

Registry hives (3)			Available bookmarks (95/0)			Values			Uninstall		
Enter text to search...			Find			Drag a column header here to group by that column					
Key name	# values	#	Timestamp	Key Name							
▼ C:	=	^	2022-10-21 11:14:04	22e0ac51							
▶ HeapLeakDetection	0										
▶ Image File Execution Options	0										
▶ Internet Explorer	9										
▶ LogonUI	8										
▶ NetworkCards	0										
▶ NetworkList	3										
▶ Products	0										
▶ UserData	0										
▶ ProfileList	4										
▶ Run	4										
▶ RunOnce	0										
▶ App Paths	0										
▶ Uninstall	0										
▶ StartMenuInternet	1										
▶ System	21										
▶ System	21										
▶ TaskCache	0										
▶ Tracing	1										
▶ VolumeInfoCache	0										
▶ Windows Portable Devices	0										
▶ Winlogon	35										
▶ Tracing	1										
▶ Uninstall	0										

Registry hives (3)			Available bookmarks (95/0)			Values			USB		
Enter text to search...			Find			Drag a column header here to group by that column					
Key name	# values	#	Timestamp	Key Name							
▼ C:	=	^									
▶ Tracing	1										
▶ Uninstall	0										
▶ C:\Windows\system32\config\SYSTEM											
▶ {10497b1b-ba51-44e5-8318-a65c837b6661}	0										
▶ {4d36e972-e325-11ce-bfc1-08002be10318}	6										
▶ {53f56307-b6bf-11d0-94f2-00a0c91efb8b}	0										
▶ {6bdd1fc6-810f-11d0-bec7-08002be2092f}	6										
▶ AppCompatCache	3										
▶ bam	7										
▶ Devices	0										
▶ ComputerName	2										
▶ CrashControl	11										
▶ DeviceClasses	0										
▶ Environment	21										
▶ EventLog	13										
▶ FilesNotToSnapshot	10										
▶ FileSystem	41										
▶ FirewallPolicy	4										
▶ Interfaces	0										
▶ Memory Management	16										
▶ MountedDevices	8										
▶ NetworkSetup2	0										
▶ PrefetchParameters	3										
▶ RDP-Tcp	85										
▶ SafeBoot	1										
▶ Services	0										
▶ Shares	0										
▶ Terminal Server	15										
▶ TimeZoneInformation	10										
▶ USB	0										

Timestamp	Key Name	Serial Number	
2024-01-09 00:59:27	ROOT_HUB30	4&1	0&0
2024-01-09 00:59:27	ROOT_HUB30	4&2	0&0
2023-12-27 05:03:54	VID_0000&PID_0002	5&1	0&3
2023-10-17 10:57:40	VID_03FD&PID_0008	5&1	0&1
2023-10-31 09:45:24	VID_03FD&PID_0008	5&1	0&3
2023-10-17 10:54:51	VID_03FD&PID_0013	5&1	0&1
2023-10-31 09:37:50	VID_03FD&PID_0013	5&1	0&3
2023-12-21 15:36:47	VID_0403&PID_6001	800	
2024-01-07 01:14:04	VID_04E8&PID_61F5	MSF	567B8F
2024-01-09 00:59:29	VID_04F2&PID_B6FA	000	
2024-01-09 00:59:30	VID_04F2&PID_B6FA&M_I_00	6&1	0&000
2023-12-20 14:40:09	VID_04FE&PID_0021	5&1	0&1
2023-12-22 07:37:46	VID_04FE&PID_0021	5&1	0&3
2023-12-20 14:40:09	VID_04FE&PID_0021&M_I_00	6&2	0&0000
2023-12-22 07:37:46	VID_04FE&PID_0021&M_I_00	6&6	0&0000
2023-12-20 14:40:09	VID_04FE&PID_0021&M_I_01	6&2	0&0000
2023-12-22 07:37:46	VID_04FE&PID_0021&M_I_01	6&6	0&0001
2023-12-20 14:40:09	VID_04FE&PID_0021&M_I_02	6&2	0&0000
2023-12-22 07:37:46	VID_04FE&PID_0021&M_I_02	6&6	0&0002

# Registry Explorer + RECcmd

- 확인하고자 하는 경로를 찾기 어렵다면 검색 기능을 활용하자.
- CTRL + F 를 통해 찾을 수도 있지만, 왼쪽 창에서 아래 사진과 같이 UserAssist 를 입력하여 바로 접근할 수 있다.

The screenshot shows the Registry Explorer v2.0.0.0 interface. The left pane displays the registry tree with 'UserAssist' selected under 'HKEY\_CURRENT\_USER'. The right pane shows the 'UserAssist' values, which are grouped by 'Program Name'. The table below represents the data shown in the right pane.

Program Name	Run Counter	Focus Count	Focus Time	Last Executed
UEME_CTLSESSION	0	0	0d, 0h, 00m, 00s	
UEME_CTLSESSION	0	0	0d, 0h, 00m, 00s	
UEME_CTLSESSION	0	0	0d, 0h, 00m, 00s	
UEME_CTLSESSION	0	0	0d, 0h, 00m, 00s	
UEME_CTLCUACount:ctor	0	0	0d, 0h, 00m, 00s	
UEME_CTLSESSION	450	4528	1d, 22h, 10m, 26s	
Microsoft.WindowsCalcu r_8wekyb3d8bbwe!App	0	0	0d, 0h, 00m, 00s	2024-01-05 09:59:33
Microsoft.Paint_8wekyb3d8 bbwe!App	3	3	0d, 0h, 00m, 23s	2024-01-17 08:20:42
Microsoft.WindowsNotepad _8wekyb3d8bbwe!App	35	167	0d, 0h, 40m, 38s	2024-01-17 16:25:44

# Registry Explorer + RECcmd

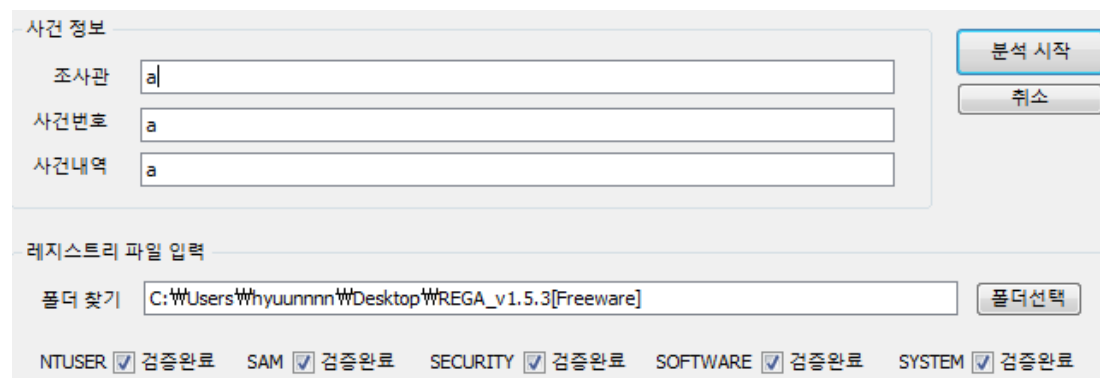
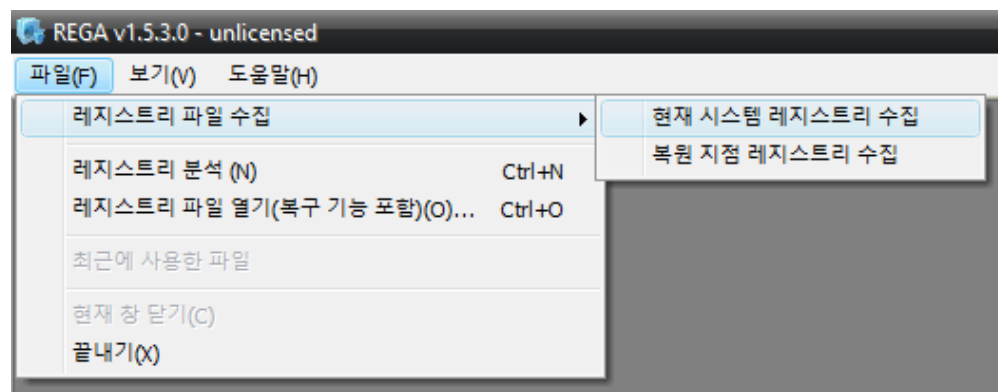
- Registry Explorer의 CLI 버전
- BatchExamples 폴더에 어떤 경로에 있는 레지스트리 데이터 수집할지 정리되어 있는 파일들이 존재한다.
  - 이를 활용하여 아래와 같은 명령어로 사용할 수 있다.
- --nl 은 트랜잭션 로그를 사용할 것인지 설정하는 옵션이다.
- --csv . 은 현재 경로에 csv 파일로 저장하는 옵션이다.

```
C:\Users\hyuunnn\Desktop\RECcmd>RECcmd.exe -d C:\Users\hyuunnn\Desktop\registry --bn "BatchExamples\Kroll_Batch.reb" --nl true --csv .  
RECcmd version 2.0.0.0  
  
Author: Eric Zimmerman (saericzimmerman@gmail.com)  
https://github.com/EricZimmerman/RECcmd  
  
Note: Enclose all strings containing spaces (and all RegEx) with double quotes  
  
Command line: -d C:\Users\hyuunnn\Desktop\registry --bn BatchExamples\Kroll_Batch.reb --nl true --csv .
```

```
RECcmd.exe -d 디렉토리명 --bn "BatchExamples\Kroll_Batch.reb" --nl true --csv .
```












- 고려대학교 DFRC와 4&6Tech에서 만든 레지스트리 분석 도구<sup>1</sup>



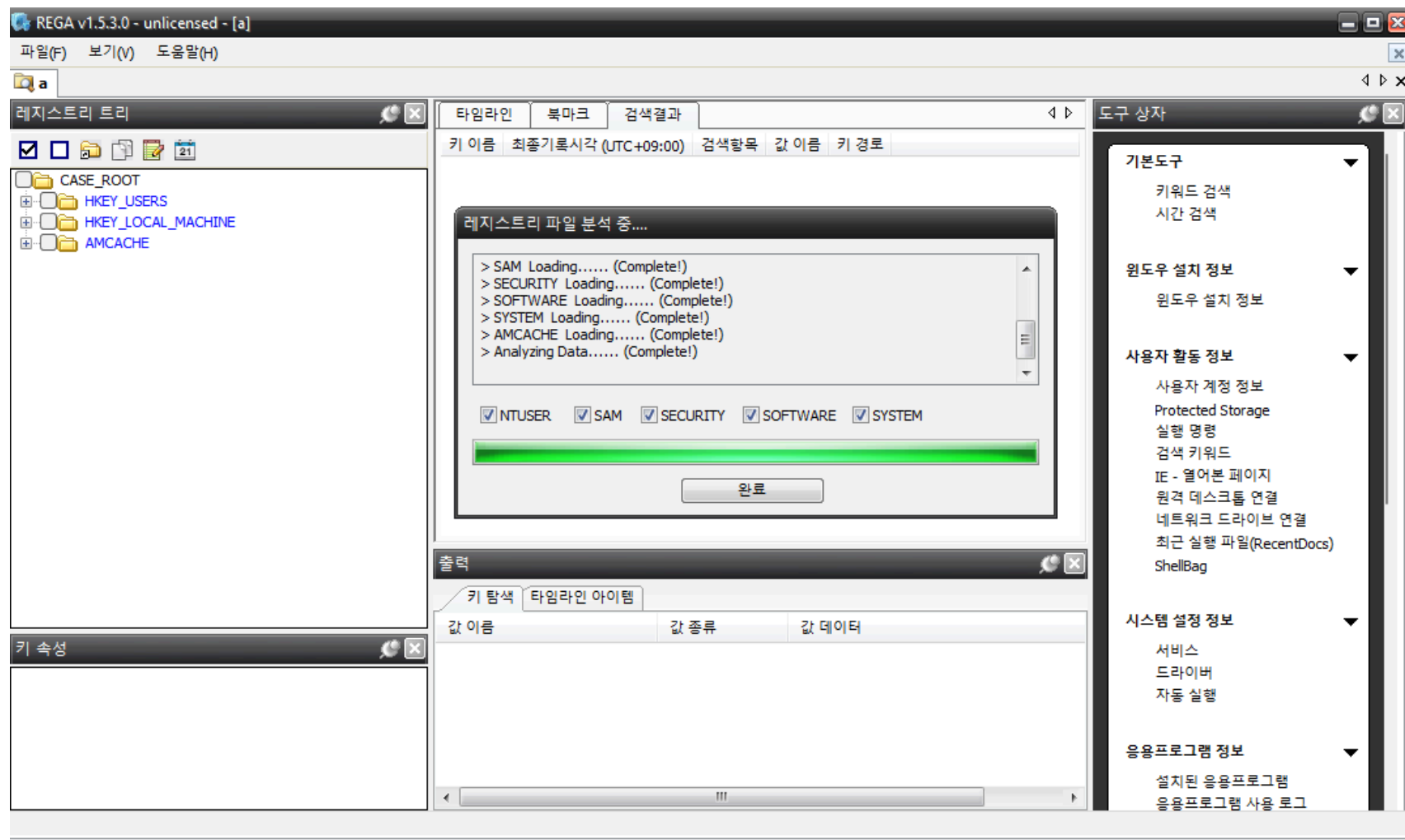
- 레지스트리 파일 저장 → 파일 → 레지스트리 분석 → 레지스트리 폴더 선택 → 분석 시작

<sup>1</sup> [http://forensic.korea.ac.kr/DFWiki/index.php/REGA\(Registry\\_Analyzer\)](http://forensic.korea.ac.kr/DFWiki/index.php/REGA(Registry_Analyzer))

- 추출된 레지스트리 파일명을 보면 NTUSER.DAT, USRCLASS.DAT 파일 이름 앞에 사용자명. 이 추가되어 있다.
- 위의 규칙을 따르지 않으면 REGA에서 인식하지 못한다.
- FTK Imager와 같은 도구에서 수동으로 추출한다면 위의 규칙을 숙지하고 있어야 한다.

 SETUPAPI	2024-01-23 오전 2:27	파일 폴더	
 Amcache.hve	2024-01-23 오전 2:30	HVE 파일	6,912KB
 COMPONENTS	2024-01-23 오전 2:30	파일	47,616KB
 DEFAULT	2024-01-23 오전 2:30	파일	768KB
 Default User.NTUSER.DAT	2024-01-23 오전 2:30	DAT 파일	256KB
 Default User.USRCLASS.DAT	2024-01-23 오전 2:30	DAT 파일	8KB
 Default.NTUSER.DAT	2024-01-23 오전 2:30	DAT 파일	256KB
 Default.USRCLASS.DAT	2024-01-23 오전 2:30	DAT 파일	8KB
 hyuunnnn.NTUSER.DAT	2024-01-23 오전 2:30	DAT 파일	4,352KB
 hyuunnnn.USRCLASS.DAT	2024-01-23 오전 2:30	DAT 파일	4,096KB

- 오른쪽 도구 상자에 있는 버튼들을 클릭하여 결과 확인 가능



# Glossy

- `pip install -r requirements.txt` 라이브러리 설치 → 관리자 권한으로 터미널 실행 → `src` 폴더까지 들어가서 `python main.py` 실행

```
C:\> 관리자: 명령 프롬프트 - python main.py
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Users\hyuunnn\Desktop\glossy-master\src

C:\Users\hyuunnn\Desktop\glossy-master\src>python main.py
Bottle v0.12.25 server starting up (using TornadoServer())...
Listening on http://127.0.0.1:9494/
Hit Ctrl-C to quit.
```

- `localhost:9494` 접속

- Add Local EVTX → Index EVTX 누르면 현재 사용 중인 PC의 이벤트 로그 분석 수행
- 이벤트 로그에 존재하는 의미 있는 로그들을 웹 환경에서 확인 가능
- 방대한, 최신의 이벤트 로그들을 모두 포함하진 않는다.

Glossy

Collect EVTX

Add Local(windir) EVTX

Upload EVTX

Index EVTX

Reset

Show All Events

System

Account

Application

Hardware

394 Sources  
등록된 EVTX 파일 수

394 / 394 Status  
(처리된 EVTX 파일 수 / 분석대상 EVTX 파일 수)

440,286 Events  
로딩된 모든 이벤트 건수

EVTX 파일 목록

☒ 모두 선택 ☐ 모두 선택해제

파일 ID	파일명	삭제
<input type="checkbox"/> 007872dd36731206d40cc572fab8d4be	c:\windows\system32\winevt\logs\Microsoft-Windows-CloudRestoreLauncher%4Operational.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 00a73f13a7cb0dd7b1722e3b934cb3aa	c:\windows\system32\winevt\logs\Microsoft-Windows-ModernDeployment-Diagnostics-Provider%4Autopilot.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 014c79badd5b0e53d702a68655a0a94f	c:\windows\system32\winevt\logs\Microsoft-Windows-TZSync%4Operational.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 018c9e6f1ee69bd8599c9d06d7833230	c:\windows\system32\winevt\logs\Microsoft-Windows-CodeIntegrity%4Operational.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 020c9a41d6066c5ddcd462213d147da5	c:\windows\system32\winevt\logs\Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Operational.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 02dcd909fa65bb6046100bcc2b4ade44	c:\windows\system32\winevt\logs\Microsoft-Windows-DeviceGuard%4Operational.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 03a01ea647a22f64d67f521d1e22bb7c	c:\windows\system32\winevt\logs\Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 03cb90e15d4d457e58845b592997dd38	c:\windows\system32\winevt\logs\Microsoft-Windows-Windows Firewall With Advanced Security%4ConnectionSecurity.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 042e295c7b241d075004bd18bc802ba9	c:\windows\system32\winevt\logs\Microsoft-Windows-Storage-Tiering%4Admin.evtx	<input type="button" value="삭제"/>
<input type="checkbox"/> 049b085e757eb1fd54cd348167f8c919	c:\windows\system32\winevt\logs\Microsoft-Windows-NetworkLocationWizard%4Operational.evtx	<input type="button" value="삭제"/>

Showing 1 to 10 of 394 entries

Previous 1 2 3 4 5 ... 40 Next

# EvtxECmd

- 이벤트 로그 분석에 자주 사용되는 CLI 도구
- 분석 결과를 CSV 파일로 추출한 후 Excel과 같은 프로그램으로 분석
  - Excel의 필터 기능을 사용하면 분석에 유의미한 결과 제공
- 동작 방식은 [Maps](#)에 있는 파일들을 활용하여 Provider와 EventID가 매칭되는 로그들을 CSV로 저장
  - 추가되지 않은 유의미한 로그가 있다면 `map` 파일을 만들어서 기여해보자.

```
C:\Users\hyuunnn\Desktop\EvtxEcmd>EvtxECmd.exe -d %SystemRoot%\System32\winevt\Logs --csv .  
EvtxECmd version 1.5.0.0  
  
Author: Eric Zimmerman (saericzimmerman@gmail.com)  
https://github.com/EricZimmerman/evtX  
  
Command line: -d C:\Windows\System32\winevt\Logs --csv .  
  
CSV output will be saved to .\20240116153643_EvtxECmd_Output.csv
```

```
EvtxECmd.exe -d %SystemRoot%\System32\winevt\Logs --csv .
```

# EvtxCmd

- Excel을 사용하는 방법

## CSV 파일 분석 꿀팁

- 첫 번째 라인 클릭 (사진 왼쪽에 1을 누르면 된다.) → 홈 → 정렬 및 필터 → 필터 클릭
  - 각 카테고리에 원하는 데이터만 볼 수 있게 필터링하거나 정렬하는 방법을 활용해보자.

	A	B	C	D	E	F	G	H	I	J
1	TimeStamp(UTC+9)	USN	File/Dir	FullPath	EventIn	Source	FileAttr	Carving	FileRef	ParentFileReferenceNumber
2	2024-01-03 16:33	88	shell32.dll	WWindow:File_Close	Normal		Archive		0x000100(0x000100000000446B	
3	2024-01-03 16:33	176	psapi.dll	WWindow:File_Close	Normal		Archive		0x000100(0x0001000000001FE3	
4	2024-01-03 16:33	256	setupapi.c	WWindow:File_Close	Normal		Archive		0x000100(0x0001000000004418	

- 타임스탬프 설정
  - Office Excel의 경우 타임스탬프로 보여주는 값이 미흡하다.
  - A 클릭 (A열 전체 드래그) → 오른쪽 클릭 → 셀 서식 → 사용자 지정 → 아래 사진과 같이 세팅

	A
1	TimeStamp(UTC+9)
2	2024-01-03 16:33:11
3	2024-01-03 16:33:11
4	2024-01-03 16:33:11
5	2024-01-03 16:33:11
6	2024-01-03 16:33:11

회계  
날짜  
시간  
백분율  
분수  
지수  
텍스트  
기타  
사용자 지정

형식(I):  
yyyy-mm-dd hh:mm:ss  
h:mm:ss AM/PM  
h:mm  
h:mm:ss  
h"시" mm"분"  
h"시" mm"분" ss"초"  
yyyy-mm-dd h:mm

# EvtxECmd

- Timeline Explorer
  - 중복되는 내용들을 그룹핑하여 로그를 볼 수 있다.

20240116153350\_EvtxECmd\_Output.csv

Map Description ▾							
	Line	Tag	Record Number	Event Record Id	Event Id	Time Created	Level
⌵	=	<input checked="" type="checkbox"/>	=	=	=	=	Info
▸ Map Description: Pipeline executed (Count: 12)							
▸ Map Description: Performing Create VHD (Count: 12)							
▸ Map Description: Performance summary for Storport Device (Count: 2,834)							
▸ Map Description: Path of executed program (Count: 985)							
▾ Map Description: OS was started (Count: 60)							
	381135	<input type="checkbox"/>	85815	85815	12	2023-10-04 08:58:51	Info
	382128	<input type="checkbox"/>	86808	86808	12	2023-10-07 09:09:16	Info
	382675	<input type="checkbox"/>	87355	87355	12	2023-10-09 22:39:18	Info
	383626	<input type="checkbox"/>	88306	88306	12	2023-10-12 00:08:16	Info
	384827	<input type="checkbox"/>	89507	89507	12	2023-10-16 06:37:55	Info
	385107	<input type="checkbox"/>	89787	89787	12	2023-10-16 09:03:44	Info
	385299	<input type="checkbox"/>	89979	89979	12	2023-10-16 09:05:09	Info



# 파일 시스템 분석

- \$MFT<sup>1</sup> - \ \$MFT
  - Master File Table의 약자이며, 볼륨에 존재하는 모든 파일과 폴더들의 정보를 가지고 있는 테이블
  - 모든 파일 및 폴더마다 하나 이상의 MFT 엔트리가 할당된다.<sup>2</sup>
    - 여러 개의 MFT 엔트리가 모여서 MFT 영역이 만들어진다.
  - 생성될 파일 수를 예측할 수 없기 때문에 일정 크기를 할당한 후 파일 수가 늘어나 할당된 MFT 영역을 초과할 경우 데이터 영역의 일정 부분을 MFT 영역으로 추가 할당하여 사용한다.<sup>2</sup>

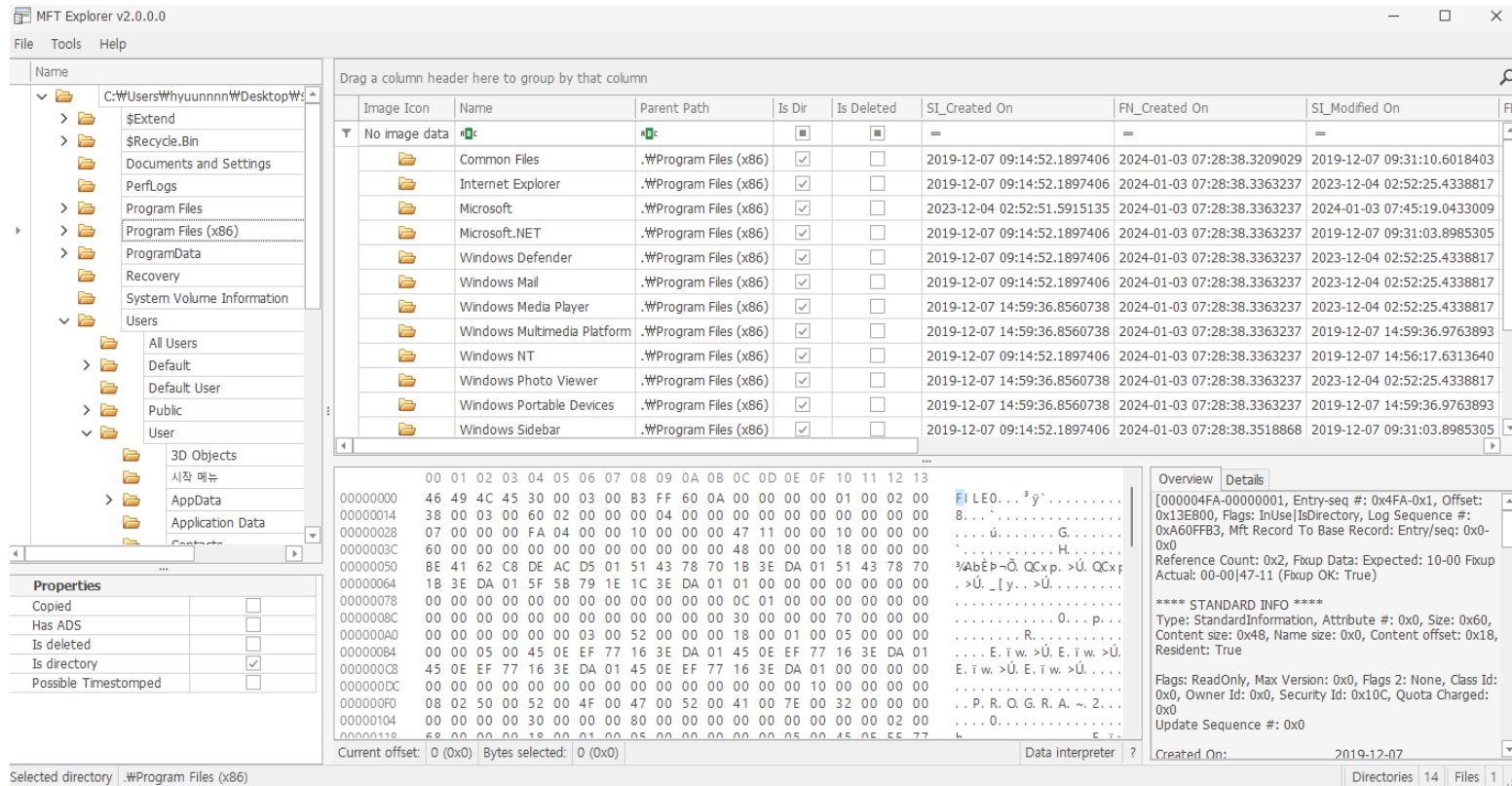
---

<sup>1</sup> <http://forensic.korea.ac.kr/DFWIKI/index.php/메타데이터/NTFS>

<sup>2</sup> <http://forensic-proof.com/archives/470>

# 파일 시스템 분석

- \$MFT<sup>1</sup> - \ \$MFT
  - MFTExplorer 또는 MFT\_Browser 도구를 사용하여 실습 가능



- 폴더 및 파일 디렉토리를 확인할 수 있다. → FTK Imager도 \$MFT 를 파싱하여 보여준 것이다.

# 파일 시스템 분석

- `$UsnJrnl:\$J`<sup>1</sup> - `\$Extend\UsnJrnl\$J`
  - NTFS의 메타데이터를 구성하는 파일, 파일시스템의 모든 파일 및 디렉터리의 변경 사항을 기록
  - `$J` 는 `$UsnJrnl` 안에 존재하는 저널 데이터이며, 해당 파일을 추출
- `$LogFile`<sup>2</sup> - `\$LogFile`
  - NTFS 트랜잭션 로그 파일, 운영체제가 비정상적으로 종료된 경우 롤백을 수행할 때 해당 파일을 활용
- 위 파일들을 분석해주는 다양한 도구들이 있지만 그 중에서 **NTFS Log Tracker**를 사용 예정
  - 분산되어 있는 데이터( `$MFT` , `$J` , `$LogFile` )들을 모아서 분석한다고 생각하면 된다.
- 시간이 된다면 **jschicht**의 도구들도 사용해 보는 것을 추천한다.
  - 파일 시스템 카빙 도구가 있다. ( `UsnJrnlCarver` , `MftCarver` 등) - [Blog 정리](#)

<sup>1</sup> [http://forensic.korea.ac.kr/DFWIKI/index.php/로그\\_%26\\_저널\\_분석/NTFS#.24UsnJrnl](http://forensic.korea.ac.kr/DFWIKI/index.php/로그_%26_저널_분석/NTFS#.24UsnJrnl)

<sup>2</sup> [http://forensic.korea.ac.kr/DFWIKI/index.php/로그\\_%26\\_저널\\_분석/NTFS#.24LogFile](http://forensic.korea.ac.kr/DFWIKI/index.php/로그_%26_저널_분석/NTFS#.24LogFile)

# NTFS Log Tracker

Basic data partition (3) [19787MB]

NONAME [NTFS]

[orphan]

[root]

\$BadClus

\$Bitmap

\$Extend

\$Recycle.Bin

\$Secure

\$UpCase

Documents and Settings

PerfLogs

Program Files

Program Files (x86)

ProgramData

Recovery

System Volume Information

Users

Windows

[unallocated space]

Partition 4 [573MB]

Unpartitioned Space [GPT]

Program Files

Program Files (x86)

ProgramData

Recovery

System Volume Informati...

Users

Windows

\$AttrDef

\$BadClus

\$Bitmap

\$Boot

\$I30

\$LogFile

\$MFT

\$MFTMirr

1 Directory

1 Directory

1 Directory

1 Directory

1 Directory

1 Directory

1 Directory

3 Regular F...

0 Regular F...

619 Regular F...

8 Regular F...

4 NTFS Ind...

29,648 Regular F...

123,136 Regular F...

4 Regular F...

2024-01-03 오전 8:20:25

2024-01-03 오전 8:04:13

2024-01-03 오전 7:53:32

2024-01-03 오전 7:35:15

2024-01-03 오전 7:37:25

2024-01-03 오전 8:10:05

2024-01-03 오전 8:20:53

2024-01-03 오전 7:28:34

2024-01-03 오전 7:28:34

2024-01-03 오전 7:28:34

2024-01-03 오전 7:28:34

2024-01-03 오전 8:20:28

2024-01-03 오전 7:28:34

2024-01-03 오전 7:28:34

[root] 경로에서 \$LogFile, \$MFT 추출

240103.E01

EFI system partition (1) [100MB]

Microsoft reserved partition (2) [16MB]

Basic data partition (3) [19787MB]

NONAME [NTFS]

[orphan]

[root]

\$BadClus

\$Bitmap

\$Extend

\$Deleted

\$ObjId

\$Reparse

\$RmMetadata

\$UsnJrnl

Name	Size	Type	Date Modified
\$J	29,354	Alternate...	2024-01-03 오전 7:33:11
\$J.FileSlack	23	File Slack	
\$Max	1	Alternate...	2024-01-03 오전 7:33:11

[root]\\$Extend\\$UsnJrnl 클릭 후 \$J 추출

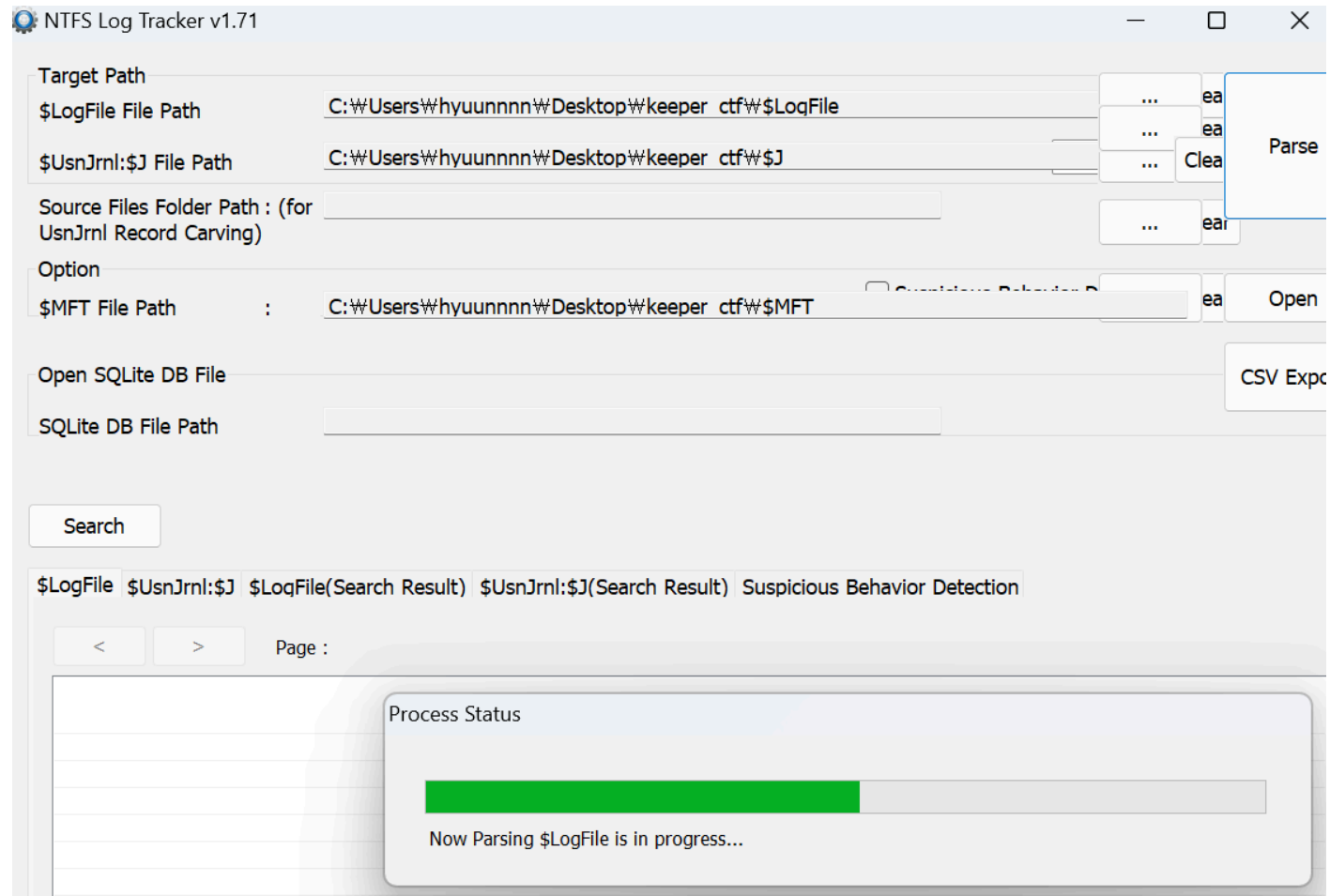
# NTFS Log Tracker 사용하기

- \$MFT, \$J, \$LogFile 이나 레지스트리 하이브 파일들은 운영체제 관련 파일들이기 때문에 탐색기를 열었을 때 보이지 않는데, 아래와 같이 설정해주면 된다.
- 파일 탐색기 옵션 검색 → 보기 클릭 → 보호된 운영 체제 파일 숨기기(권장) 체크 해제
- 나머지 옵션은 추가로 숨겨진 파일 확인이나 확장자를 수정할 때 편하기 때문에 모두 보이게 설정한다.

- ☐ 보호된 운영 체제 파일 숨기기(권장)
- ☒ 빈 드라이브 숨기기
- ☒ 상태 표시줄 표시
- ☒ 숨김 파일 및 폴더
  - ☐ 숨김 파일, 폴더 또는 드라이브 표시 안 함
  - ☒ 숨김 파일, 폴더 및 드라이브 표시
- ☐ 아이콘은 항상 표시하고 미리 보기는 표시하지 않음
- ☐ 알려진 파일 형식의 파일 확장명 숨기기

# NTFS Log Tracker 사용하기

- 추출한 3개의 파일을 올린 후 Parse 버튼 클릭 → SQLite 파일명 및 경로는 아무 곳이나 상관 없음



# NTFS Log Tracker 사용하기

- 분석이 완료되었다면 아래와 같은 결과가 보이는데, 더욱 편하고 의미있는 분석을 하기 위해 CSV 추출  
CSV Export → 경로 설정 후 확인

Open SQLite DB File

SQLite DB File Path

Search

CSV Export

\$LogFile \$UsnJrnl:\$J \$LogFile(Search Result) \$UsnJrnl:\$J(Search Result) Suspicious Behavior Detection










< > Page : ( 1 / 1 )

LSN	EventTime(UT...	Event	Detail	File/Directory Name	Full Path(from \$MFT)
181081698		Writing Content of No...	Data Runs(in Volume)...		
181082147	2024-01-03 17...	Directory Creation		th	
181082456	2024-01-03 17...	File Creation		FileSync.LocalizedRes...	
181082764		Writing Content of No...	Data Runs(in Volume)...	FileSync.LocalizedRes...	
181083208	2024-01-03 17...	Directory Creation		ti	
181083517	2024-01-03 17...	File Creation		FileSync.LocalizedRes...	
181083822		Writing Content of No...	Data Runs(in Volume)...	FileSync.LocalizedRes...	
181084287	2024-01-03 17...	Directory Creation		tk-TM	
181084599	2024-01-03 17...	File Creation		FileSync.LocalizedRes...	
181084908		Writing Content of No...	Data Runs(in Volume)...	FileSync.LocalizedRes...	
181085358	2024-01-03 17...	Directory Creation		tn-ZA	
181085670	2024-01-03 17...	File Creation		FileSync.LocalizedRes...	
181085975		Writing Content of No...	Data Runs(in Volume)...	FileSync.LocalizedRes...	

LogFile Record Count : 10408 \$UsnJrnl Record Count : 227969 id by Junghoon Oh( blueangel12

# NTFS Log Tracker 사용하기

- 지금까지의 모든 단계를 따라왔다면 아래 사진과 같은 파일들을 확인할 수 있다.
- 레지스트리 및 CSV 파일들을 분석하여 랜섬웨어의 행위를 분석하고 답을 찾아보자.

 \$J	2024-01-03 오후 4:33	시스템 파일	29,354KB
 \$LogFile	2024-01-03 오후 4:28	시스템 파일	29,648KB
 \$MFT	2024-01-03 오후 4:28	시스템 파일	123,136KB
 NLT_LogFile_2024-01-14 00-22-27.csv	2024-01-14 오전 12:22	Microsoft Excel 싼표...	2,522KB
 NLT_LogFile_Search_2024-01-14 00-22-27.csv	2024-01-14 오전 12:22	Microsoft Excel 싼표...	1KB
 NLT_Suspicious_Behavior_Detection_2024-01-14 00-22-27.csv	2024-01-14 오전 12:22	Microsoft Excel 싼표...	1KB
 NLT_UsnJrnl_2024-01-14 00-22-27.csv	2024-01-14 오전 12:22	Microsoft Excel 싼표...	51,069KB
 NLT_UsnJrnl_Search_2024-01-14 00-22-27.csv	2024-01-14 오전 12:22	Microsoft Excel 싼표...	1KB
 test_2024-01-14 00-17-35.db	2024-01-14 오전 12:17	Data Base File	131,035KB

- Excel 프로그램, [Timeline Explorer](#) 등 자신에게 편리한 프로그램으로 CSV 분석



# CSV 파일 분석 꿀팁

- 랜섬웨어가 어떤 행위를 할까?
  - 랜섬웨어는 특정 파일들을 암호화시킬 때 특정 확장자로 변하지 않던가?
  - 랜섬웨어 동작이 끝나면 무엇을 하지?
    - 랜섬노트가 생성되지 않나?
- 이러한 행위들을 생각하고, 분석하여 랜섬웨어의 전체적인 동작 흐름을 분석해보자.

# 주의사항

- 랜섬웨어를 실행하지 않게 조심하자. - 랜섬웨어를 사용하는 시나리오이기 때문
- 랜섬웨어 파일을 추출했을 때 백신이 켜져있다면 자동으로 삭제되므로 백신을 켜두자.

## 바이러스 및 위협 방지 설정

Microsoft Defender 바이러스 백신에 대한 바이러스 및 위협 방지 설정을 보고 업데이트할 수 있습니다.

### 실시간 보호 기능

맬웨어를 찾고 디바이스에서 설치되거나 실행하는 것을 방지합니다. 이 설정을 잠시 동안 끌 수 있습니다. 그러면 자동으로 다시 켜집니다.



컴

# 과제 - KEEPER CTF IR-2 풀어보기

피해자는 윈도우 PC를 사용할 때 잦은 알림이 번거롭다고 느껴, 디펜더를 비활성화하는 프로그램을 항상 사용한다고 한다.

또한 피해자에게 들은 바로는 랜섬웨어가 감염되기 전에 컴퓨터가 이상한 행위를 했었다고 한다.

원인을 찾아내고, 어떤 경로로 유입되었는지 분석하라.

다운로드 유입 URL, 다운로드 받은 악성 파일, 다운로드 받은 악성 파일이 실행된 시간을 답으로 입력해야 한다.

파일명은 소문자로 입력, 띄어쓰기는 언더바(\_) 처리, 타임스탬프는 한국 시간인 UTC+9를 따르며, ISO 8601 표준에 의해 날짜와 시간 사이에 T 문자를 입력한다.

ex: KEEPER{[https://www.example.com/\\_asdf.asd\\_2024-12-23T12:34:56](https://www.example.com/_asdf.asd_2024-12-23T12:34:56)}

# 과제 - KEEPER CTF IR-2 풀어보기

- 다운로드 유입 URL? 다운로드 받은 악성 파일?
  - 다운로드를 보통 어디서 받을까? - 인터넷에서 다운로드 받는다.
  - 방문한 웹 사이트, 다운로드 파일 등을 분석해야 한다.
  - 피해자의 PC에는 어떤 브라우저들이 설치되어 있을까?
    - 브라우저마다 관련 파일들을 저장하는 경로가 다르다.  
→ 구글링으로 해당 경로를 찾은 후 추출하여 분석해보자!
- [hindsight](#) 도구 활용해보기
- 어떻게 해야 하는지 모르겠다면 디스코드에 업로드한 라이트업을 참고하자.

# 참고자료

- [Digging Up the Past: Windows Registry Forensics Revisited - mandiant](#)
- [Windows registry file format specification - msuhanov](#)
- [Exploring Registry Explorer](#)
- [Introducing AboutDFIR's Registry Explorer/RECmd Guide](#)
- [NTFS – MFT 엔트리 소개](#)
- [NTFS – MFT 엔트리 구조](#)
- [MFT\(Master File Table\) 구조](#)
- [NTFS - wikipedia](#)
- [Timeline Explorer Tutorial - aboutdfir](#)
- [기초부터 따라하는 디지털포렌식](#)