

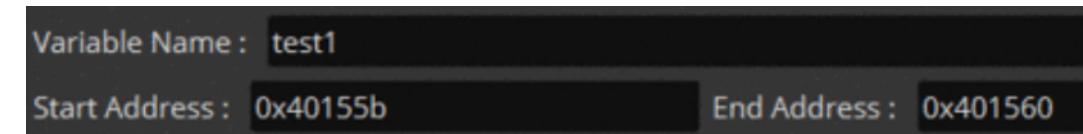
# 2주차 스터디

GUI 도구 개발 Tip + 1주차 수업 요약, 정리 + 이벤트 로그 분석

# PyQt

- C++에서 사용할 수 있는 GUI 개발 프레임워크인 Qt를 파이썬에 바인딩한 버전
- Python을 사용하여 GUI 프로그램 개발이 가능하다.
- 정교하고 세밀한 UI를 만들 수 있다. → 그만큼 시간이 많이 걸린다.

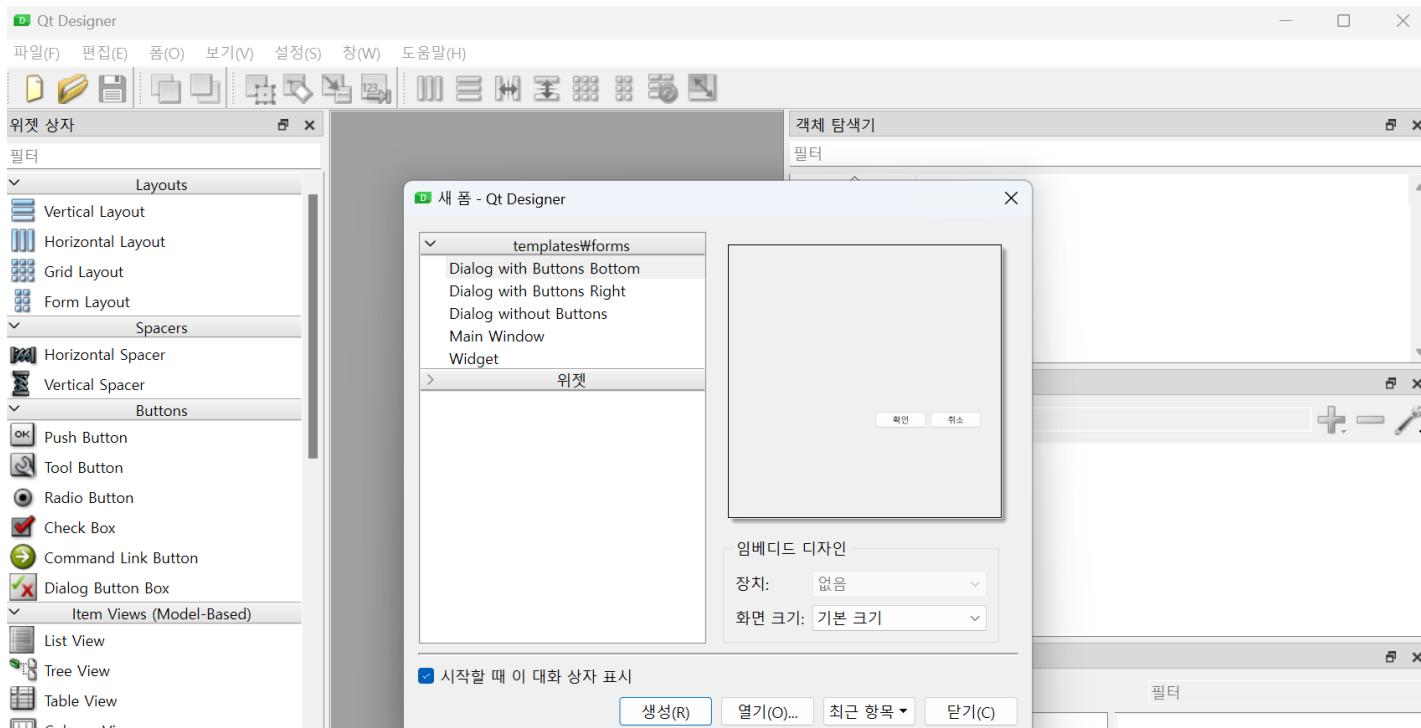
```
... 79     def _ui_init_layout(self):  
80         GL1 = QtWidgets.QGridLayout()  
81         GL1.addWidget(QtWidgets.QLabel("Variable Name : "), 0, 0)  
82         GL1.addWidget(self._variable_name, 0, 1)  
83         self.layout.addLayout(GL1)  
84  
85         GL2 = QtWidgets.QGridLayout()  
86         GL2.addWidget(QtWidgets.QLabel("Start Address : "), 0, 0)  
87         GL2.addWidget(self._start_address, 0, 1)  
88         GL2.addWidget(QtWidgets.QLabel("End Address : "), 0, 3)  
89         GL2.addWidget(self._end_address, 0, 4)  
90         self.layout.addLayout(GL2)
```



- 레이아웃 지정, 위젯 생성 및 위치 설정 등 코드가 복잡해진다. (자세한 사진 코드는 [Link](#) 참고)

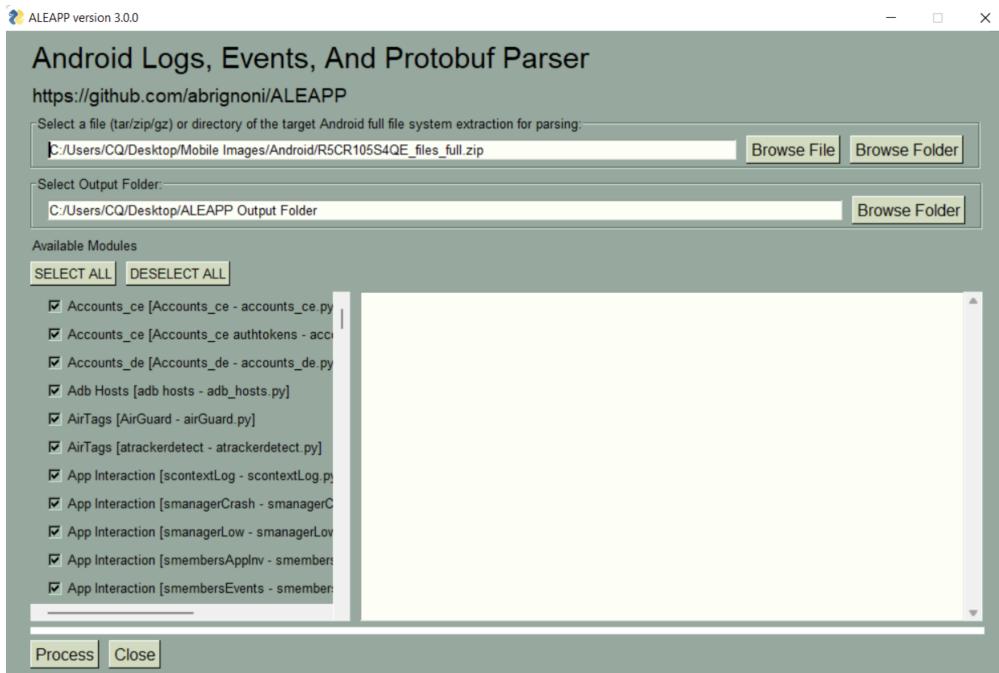
# PyQt

- Qt Designer를 사용하여 디자인할 수 있으며, UI 결과를 python 코드로 추출할 수 있다. ([ui to py](#))
  - 복잡한 코드임은 변함없다. → 버튼마다 수행해야 하는 동작, 코드를 설정해야 한다.
- pip install PySide6 설치하면 designer.exe 파일이 포함된다.
  - %LocalAppData%\Programs\Python\Python310\Lib\site-packages\PySide6에서 확인

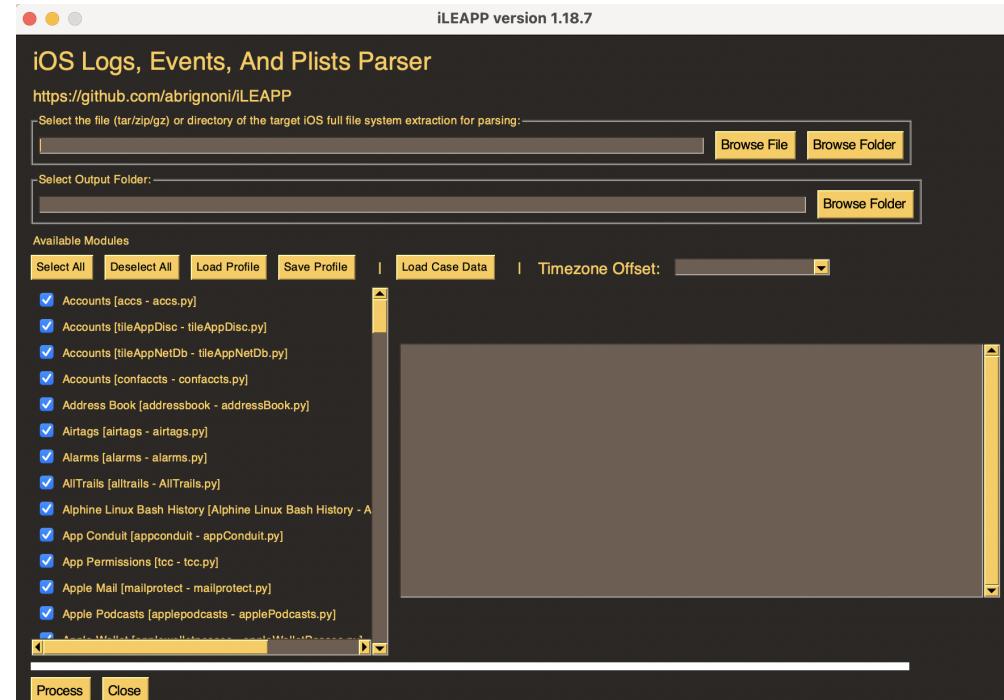


# PySimpleGUI

- 편의성을 위해 GUI 환경은 필요하지만 디자인은 신경쓰지 않는다면 추천하는 라이브러리이다.
- 간편하지만 디자인이 한정적이라는 단점이 있다.



ALEAPP



iLEAPP

# PySimpleGUI

```
import PySimpleGUI as sg

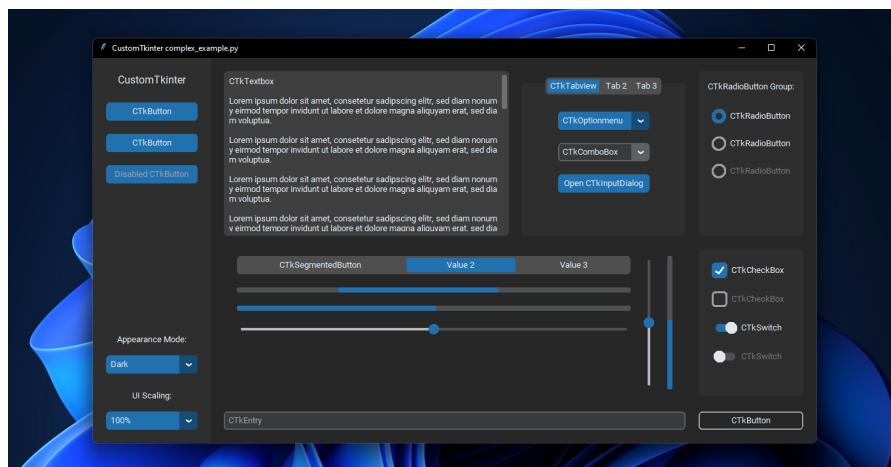
sg.theme('DarkAmber')
layout = [ [sg.Text('Some text on Row 1')],
           [sg.Text('Enter something on Row 2'), sg.InputText()],
           [sg.Button('Ok'), sg.Button('Cancel')] ]

window = sg.Window('Window Title', layout)
while True:
    event, values = window.read()
    if event == sg.WIN_CLOSED or event == 'Cancel':
        break
    print('You entered ', values[0])
```

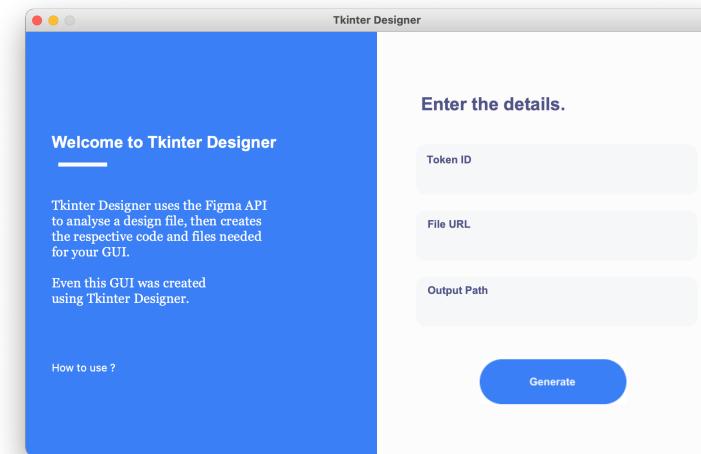


# Tkinter

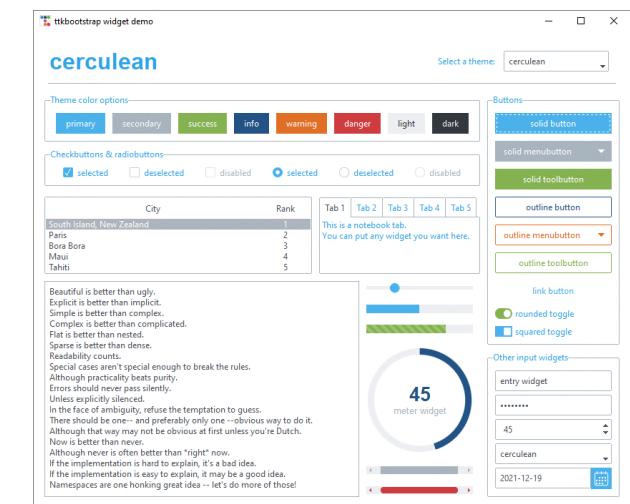
- Tk는 GUI 개발에서 사용되는 위젯들을 제공하는 툴킷
  - Tkinter는 Tk를 파이썬에서 사용하기 위해 지원하는 표준 인터페이스
- turtle 라이브러리도 tkinter를 사용하였다.
- tkinter에서 사용하기 위한 UI를 디자인할 수 있는 도구들이 존재한다. → 사용해보진 않았다.



CustomTkinter



Tkinter-Designer



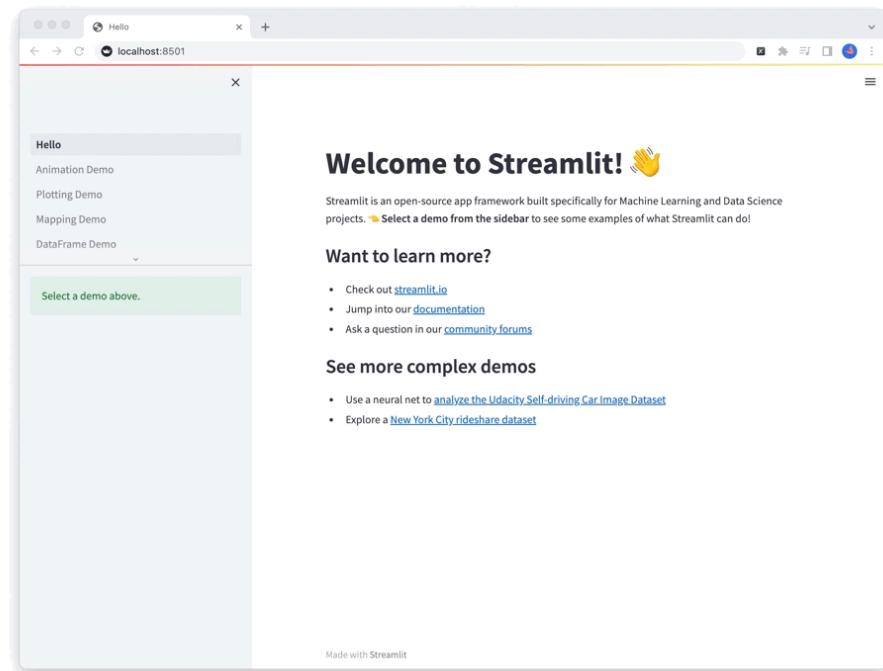
ttkbootstrap

# etc

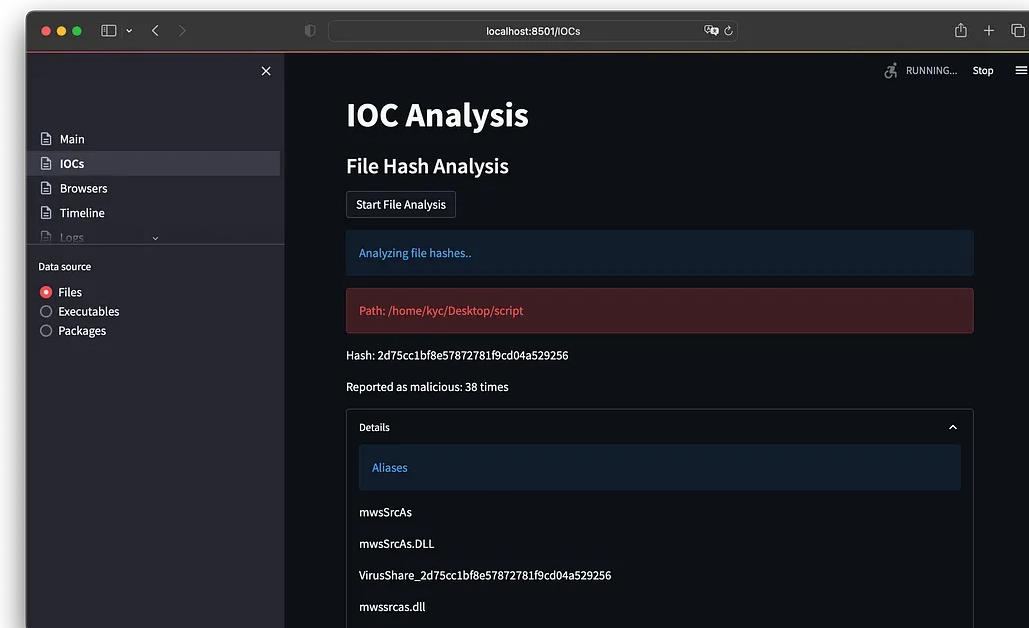
- [flet](#)
- [wxPython - Phoenix](#)
- [kivy](#)
- [textual](#)
- [pyimgui](#)
- [remi](#)
- ...

# Streamlit

- 주로 ML, Data Science에서 도출된 결과를 웹으로 보여주기 위한 용도로 사용된다.
- 웹앱을 간편하게 만들 수 있다. → 포렌식 도구 개발에 활용되진 않는 것 같다.



streamlit docs



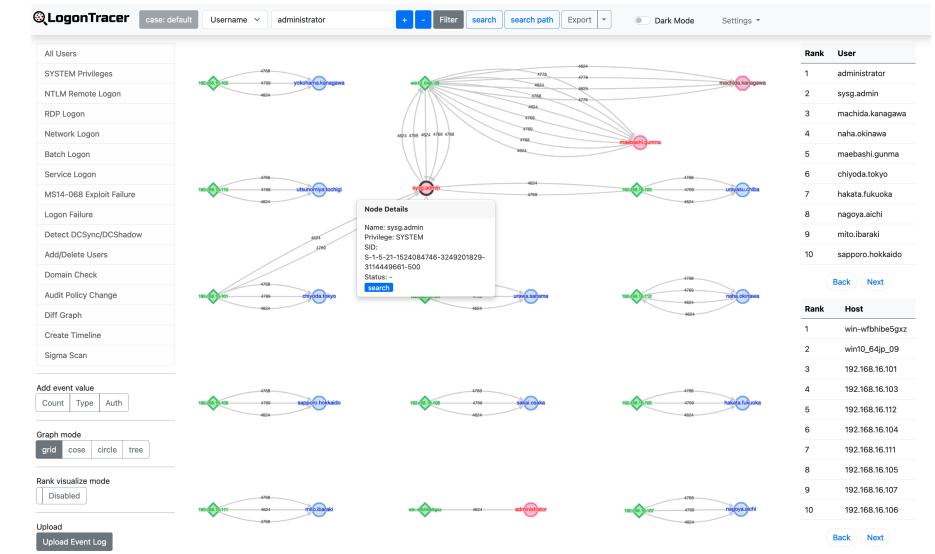
seeker

# Backend + CSS 프레임워크

- 웹 환경을 활용하면 다양한 시각화 작업이 가능하다.
  - 시각화 관련 Javascript 라이브러리가 풍부하다. ([d3.js](#), [cytoscape.js](#) ...)
- Backend ([Flask](#) ...) + CSS 프레임워크 ([Bootstrap](#), [tailwind](#), [materialize](#) ...) + 템플릿 등을 활용하여 보다 간편하게 디자인 설계 및 개발 가능 → 웹 환경에서 개발하는 것이 자유도가 높다.

The screenshot shows the Glossy application interface. On the left is a sidebar with navigation links: 'Collect EVTX' (selected), 'Add Local(windr) EVTX', 'Upload EVTX', 'Index EVTX', 'Reset', 'Show All Events', 'System', 'Account', 'Application', and 'Hardware'. The main area displays the following statistics: 324 Sources (324 / 324), 59,294 Events. Below this is a table titled 'EVTX 파일 목록' (EVTX File List) containing 324 entries. Each entry includes a checkbox, a file ID, a path, and several red '삭제' (Delete) buttons. At the bottom of the table are buttons for 'Previous', 'Next', and 'Show 10 entries'.

Glossy



LogonTracer

# Backend + CSS 프레임워크

- Flask + Bootstrap 사용

EventID	Computer	EventType	ProfileName	SSID	BSSType	CipherType	CipherAlgorithm	SystemTime
8001	DESKTOP-QQGV51B	연결 성공	aaaa	aaaa	Infrastructure	WPA2-Personal	AES-CCMP	2018-02-05T06:15:21.478810700Z
8003	DESKTOP-QQGV51B	연결 종료	aaaa	aaaa	Infrastructure	None	None	2018-02-05T09:46:39.873461700Z
8000	DESKTOP-QQGV51B	연결 시작	BoB	BoB	Infrastructure	None	None	2018-02-05T09:52:00.885486400Z
8001	DESKTOP-QQGV51B	연결 성공	BoB	BoB	Infrastructure	WPA2-Personal	AES-CCMP	2018-02-05T09:52:01.170710100Z
8003	DESKTOP-QQGV51B	연결 종료	BoB	BoB	Infrastructure	None	None	2018-02-05T14:41:45.378460300Z
8000	DESKTOP-QQGV51B	연결 시작	SK_WiFi564E	SK_WiFi564E	Infrastructure	None	None	2018-02-05T14:43:25.177754400Z
8001	DESKTOP-QQGV51B	연결 성공	SK_WiFi564E	SK_WiFi564E	Infrastructure	WPA2-Personal	AES-CCMP	2018-02-05T14:43:25.277969600Z
8003	DESKTOP-QQGV51B	연결 종료	SK_WiFi564E	SK_WiFi564E	Infrastructure	None	None	2018-02-06T01:30:30.207645900Z
8000	DESKTOP-QQGV51B	연결 시작	sunrins	sunrins	Infrastructure	None	None	2018-02-06T02:22:21.071281700Z
8001	DESKTOP-QQGV51B	연결 성공	sunrins	sunrins	Infrastructure	WPA2-Enterprise	AES-CCMP	2018-02-06T02:22:21.499345500Z

« 1 2 3 4 5 6 7 8 9 »

- 딥페이크 탐지 도구를 만든다고 가정했을 때 PC 외에 모바일 환경에서도 사용해야 한다면?
  - Flutter나 React Native가 괜찮은 방법일 수도 있다.
- 별도의 외부 프로그램이 설치되지 않은 클린한 상태에서 바로 개발한 도구를 사용하고 싶다면?
  - Powershell이 괜찮은 방법일 수도 있다. (ex: WELA, kacos2000 - Github 등)
- 결론은 상황에 맞는 적절한 방법을 선택해서 개발하면 된다.

## 참고자료

[Which Python GUI library should you use?](#)

[What is the differences between Tkinter, WxWidgets and PyQt, and PySide?](#)

[Pygtk VS PyQt VS WxPython VS Tkinter](#)

[Best Python GUI Libraries Compared! \(PyQt, Kivy, Tkinter, PySimpleGUI, WxPython & PySide\)](#)

[What is the difference between a wrapper, a binding, and a port?](#)

# 1주차 수업 요약, 정리

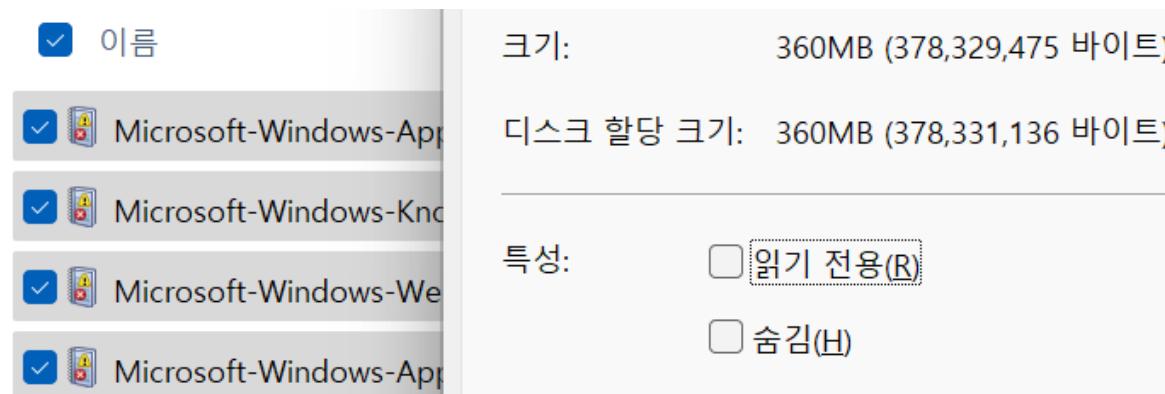
- 레지스트리는 윈도우 운영체제에서 설정과 관련된 정보를 담고있는 데이터베이스
- 최근 열람 파일, 설치 파일, 연결 기록 등을 저장하고 있다.  
→ 증거로 활용될 수 있다. (ex: 기밀 자료 유출 사건, 증거인멸 - Eraser 프로그램 설치 및 실행 기록 등)

# 이벤트 로그 분석

개념 설명, Provider & EventID, 관련 도구 설명

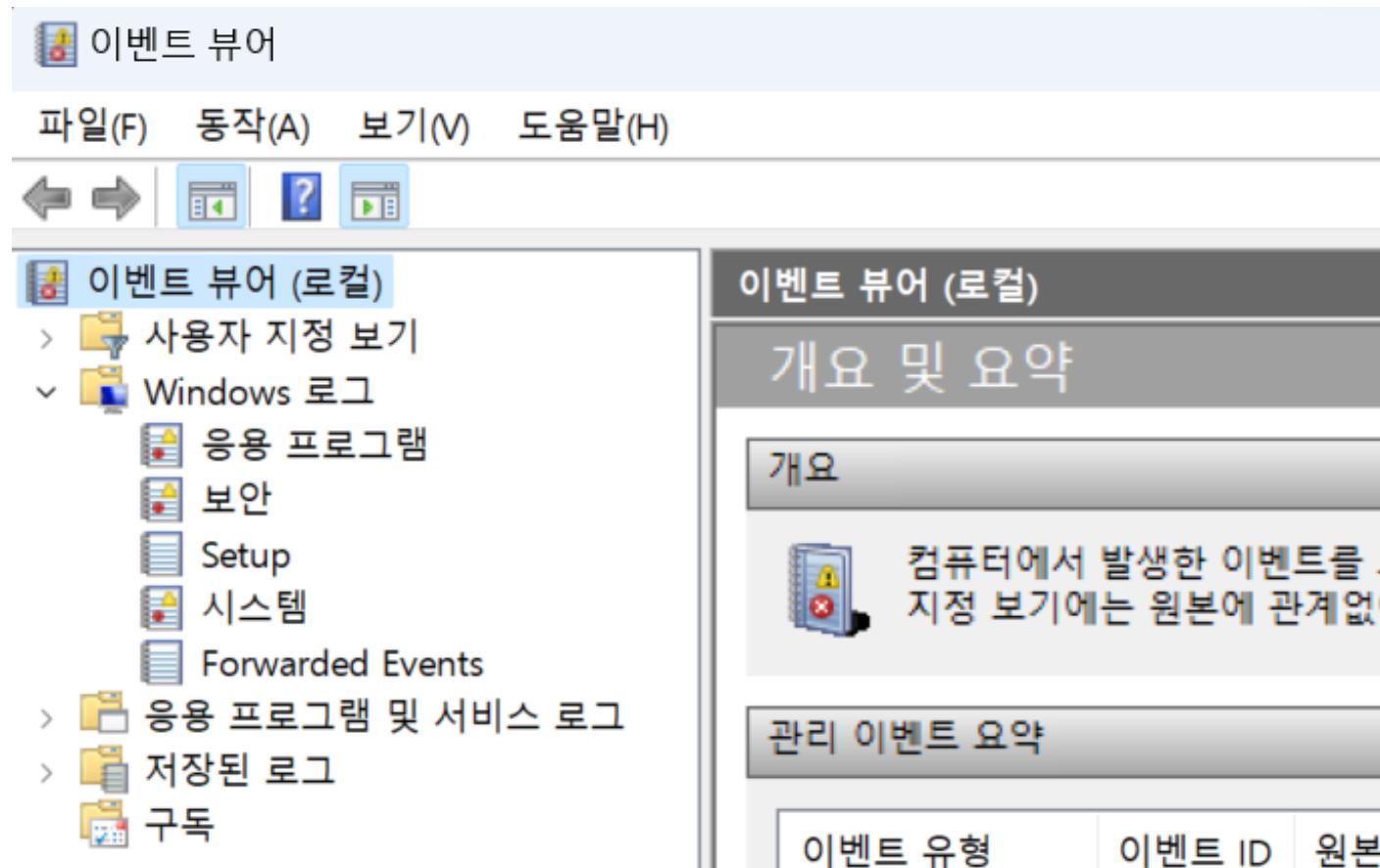
# 이벤트 로그

- 이벤트 로그란 윈도우 시스템의 모든 기록을 담고 있다. → 방대한 로그 데이터가 존재한다.
- WIFI, USB 연결 정보 등 레지스트리와 겹치는 정보들이 존재
  - 레지스트리와 비교하여 믿을만한 데이터인지 확인하기도 함
  - 레지스트리와 이벤트 로그를 같이 보면서 타임라인을 그려나간다.
- 파일 경로: **%SystemRoot%\System32\winevt\Logs**
  - 현재 사용 중인 PC의 경우 전체 파일의 용량은 362MB, 개수는 396개



# 이벤트 로그

- 이벤트 뷰어 혹은 eventvwr.msc 검색 후 프로그램 실행



# Provider & Event ID

- Provider: 이벤트 생성을 해주는 제공자의 이름
- EventID: 이벤트를 식별하기 위한 번호

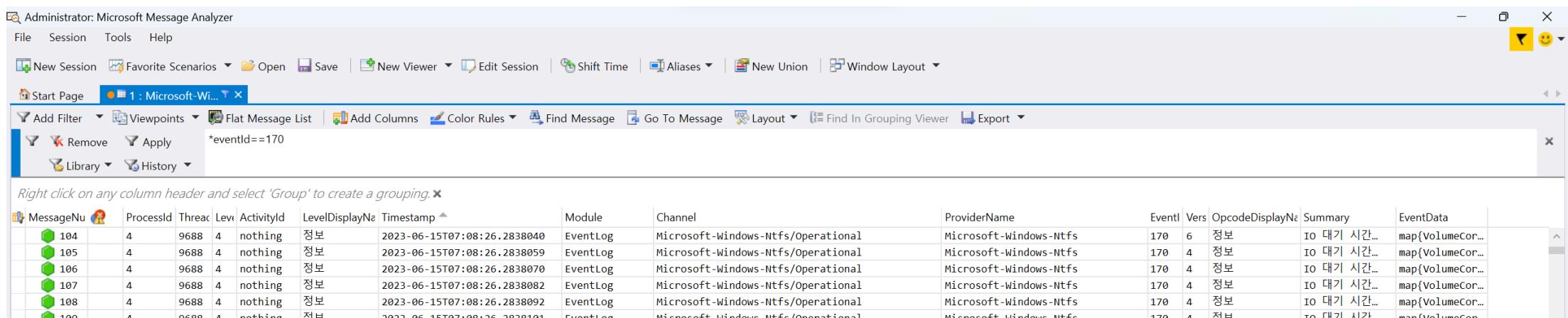
## 이벤트 로그

- Application (응용 프로그램): 사용자의 어플리케이션 이벤트 기록
  - ex:) Provider: Msilnstaller, EventID: 1034
- Security (보안): 로그인/로그아웃, 네트워크 등 보안 관련 이벤트 로그 기록
  - ex:) Provider: Microsoft-Windows-Security-Auditing, EventID: 4647
- System (시스템): 서비스 실행 여부, 시스템 에러 등 시스템 관련 이벤트 로그 기록
  - ex:) Provider: DriverFrameworks-UserMode, EventID: 10000
- 응용 프로그램 및 서비스 로그: 위 로그를 제외한 나머지 이벤트 로그

# Microsoft Message Analyzer

**Microsoft Message Analyzer** is a powerful tool used for capturing, displaying, and analyzing protocol messaging traffic, events, and other system or application messages in network troubleshooting and other diagnostic scenarios. Join Lex Thomas and Paul Long as they walk us through the new Message Analyzer interface and show us how decryption works.

- 현재는 개발 중단된 프로젝트이며, 2019. 11. 25 이후로 공식적인 배포를 하지 않는다고 발표
- 그러나 Github에 올라온 설치 파일을 통해 사용 가능 ([공식 튜토리얼](#), [설명 영상](#)은 존재함)
- 사용법이 쉽진 않다고 한다. → 그래도 찾는 사람이 많은 만큼 좋은 도구라고 생각한다.



The screenshot shows the Microsoft Message Analyzer application window. The menu bar includes File, Session, Tools, and Help. The toolbar contains buttons for New Session, Open, Save, New Viewer, Edit Session, Shift Time, Aliases, New Union, and Window Layout. The main interface has tabs for Start Page and 1 : Microsoft-Wi... The toolbar below the tabs includes Add Filter, Viewpoints, Flat Message List, Add Columns, Color Rules, Find Message, Go To Message, Layout, Find In Grouping Viewer, and Export. A search bar displays the query \*eventId=170. The bottom pane is a grid-based viewer showing captured messages. The columns are: MessageNum, ProcessId, Thread, Level, ActivityId, LevelDisplayName, Timestamp, Module, Channel, ProviderName, EventId, Vers, OpcodeDisplayName, Summary, and EventData. The data grid lists several entries, all from the EventLog module and Microsoft-Windows-Ntfs/Operational channel, with EventId 170 and Level 4 (Information).

MessageNum	ProcessId	Thread	Level	ActivityId	LevelDisplayName	Timestamp	Module	Channel	ProviderName	EventId	Vers	OpcodeDisplayName	Summary	EventData
104	4	9688	4	nothing	정보	2023-06-15T07:08:26.2838040	EventLog	Microsoft-Windows-Ntfs/Operational	Microsoft-Windows-Ntfs	170	6	정보	IO 대기 시간...	map{VolumeCor...
105	4	9688	4	nothing	정보	2023-06-15T07:08:26.2838059	EventLog	Microsoft-Windows-Ntfs/Operational	Microsoft-Windows-Ntfs	170	4	정보	IO 대기 시간...	map{VolumeCor...
106	4	9688	4	nothing	정보	2023-06-15T07:08:26.2838070	EventLog	Microsoft-Windows-Ntfs/Operational	Microsoft-Windows-Ntfs	170	4	정보	IO 대기 시간...	map{VolumeCor...
107	4	9688	4	nothing	정보	2023-06-15T07:08:26.2838082	EventLog	Microsoft-Windows-Ntfs/Operational	Microsoft-Windows-Ntfs	170	4	정보	IO 대기 시간...	map{VolumeCor...
108	4	9688	4	nothing	정보	2023-06-15T07:08:26.2838092	EventLog	Microsoft-Windows-Ntfs/Operational	Microsoft-Windows-Ntfs	170	4	정보	IO 대기 시간...	map{VolumeCor...
109	4	9688	4	nothing	정보	2023-06-15T07:08:26.2838101	EventLog	Microsoft-Windows-Ntfs/Operational	Microsoft-Windows-Ntfs	170	4	정보	IO 대기 시간...	map{VolumeCor...

# Glossy

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

```
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\hyuunnnn\Desktop\glossy-master\src

C:\Users\hyuunnnn\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

# Glossy

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

The screenshot shows the Glossy web application interface. On the left is a sidebar with navigation links: Collect EVTX, Add Local(windir) EVTX, Upload EVTX, Index EVTX (which is selected), and Reset. Below these are sections for Show All Events, System, Account, Application, and Hardware. The main content area displays the following statistics:

- Sources: 394 (등록된 EVTX 파일 수)
- Status: 394 / 394 (처리된 EVTX 파일 수 / 분석 대상 EVTX 파일 수)
- Events: 440,286 (로딩된 모든 이벤트 건수)

Below these stats is a table titled "EVTX 파일 목록" (EVTX File List) showing 10 entries out of 394:

파일 ID	파일명	삭제
007872dd36731206d40cc572fab8d4be	c:\windows\system32\winevt\logs\Microsoft-Windows-CloudRestoreLauncher%4Operational.evtx	삭제
00a73f13a7cb0dd7b1722e3b934cb3aa	c:\windows\system32\winevt\logs\Microsoft-Windows-ModernDeployment-Diagnostics-Provider%4Autopilot.evtx	삭제
014c79badd5b0e53d702a68655a0a94f	c:\windows\system32\winevt\logs\Microsoft-Windows-TZSync%4Operational.evtx	삭제
018c9e6f1ee69bd8599c9d06d7833230	c:\windows\system32\winevt\logs\Microsoft-Windows-CodeIntegrity%4Operational.evtx	삭제
020c9a41d6066c5ddcd462213d147da5	c:\windows\system32\winevt\logs\Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider%4Operational.evtx	삭제
02dc909fa65bb6046100bcc2b4ade44	c:\windows\system32\winevt\logs\Microsoft-Windows-DeviceGuard%4Operational.evtx	삭제
03a01ea647a22f64d67f521d1e22bb7c	c:\windows\system32\winevt\logs\Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant.evtx	삭제
03cb90e15d4d457e58845b592997dd38	c:\windows\system32\winevt\logs\Microsoft-Windows-Windows Firewall With Advanced Security%4ConnectionSecurity.evtx	삭제
042e295c7b241d075004bd18bc802ba9	c:\windows\system32\winevt\logs\Microsoft-Windows-Storage-Tiering%4Admin.evtx	삭제
049b085e757eb1fd54cd348167f8c919	c:\windows\system32\winevt\logs\Microsoft-Windows-NetworkLocationWizard%4Operational.evtx	삭제

At the bottom, it says "Showing 1 to 10 of 394 entries" and has navigation buttons for Previous, Next, and page numbers 1 through 40.

# EvtxEcmd

- 데이터가 클수록 프로그램이 무거워지기 때문에 중간에 죽을 수도 있다.
  - 분석 결과를 CSV 파일로 추출한 후 Excel과 같은 프로그램으로 분석하는 방법이 좋을 수도 있다.
  - CSV 파일을 분석할 때 해당 프로그램에서 제공하는 필터 기능이 분석에 편리함을 제공한다.
- 동작 방식은 [Maps](#)에 있는 파일들을 활용하여 Provider와 EventID가 매칭되는 로그들을 CSV로 저장

```
C:\Users\hyuunnnn\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 .

# EvtxECmd

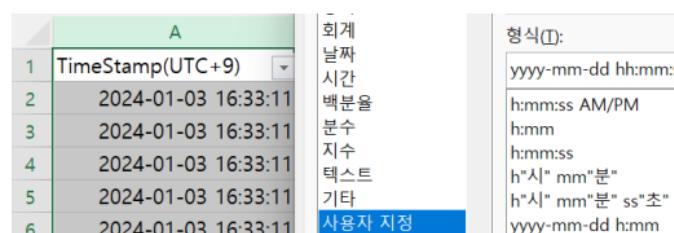
- Excel을 사용하는 방법

## 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	EventID	SourceID	FileAttri	Carving	FileRef
2	2024-01-03 16:33	88	shell32.dll	#Window.File_Close	Normal	Archive		0x0001000000000000446B	
3	2024-01-03 16:33	176	psapi.dll	#Window.File_Close	Normal	Archive		0x00010000000000001FE3	
4	2024-01-03 16:33	256	setupapi.dll	#Window.File_Close	Normal	Archive		0x00010000000000004418	

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



# 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
▼	=	■	=	=	=	=	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	□	85815	85815	12	2023-10-04 08:58:51	▲	Info
382128	□	86808	86808	12	2023-10-07 09:09:16	▲	Info
382675	□	87355	87355	12	2023-10-09 22:39:18	▲	Info
383626	□	88306	88306	12	2023-10-12 00:08:16	▲	Info
384827	□	89507	89507	12	2023-10-16 06:37:55	▲	Info
385107	□	89787	89787	12	2023-10-16 09:03:44	▲	Info
385299	□	89979	89979	12	2023-10-16 09:05:09	▲	Info

# 과제

- 간단한 이벤트 로그 분석 도구 만들어보기
  - 다양한 자료들을 찾아보면서 해당 로그가 어떤 Provider와 Event ID를 사용하는지 알아보기
    - [윈도우 이벤트 로그\(EVTX\) 분석 및 포렌식 활용방안](#)
    - [forensic-cheatsheet - plainbit](#)
    - ...
  - 파이썬 라이브러리 활용
    - [libevtx-python](#), [python-evtx](#), [pyevtx-rs](#) 등

# 참고자료

[Windows Event Logs 분석 - forensic-artifacts](#)

[forensic-cheatsheet - plainbit](#)

[Timeline Explorer Tutorial - aboutdfir](#)

[기초부터 따라하는 디지털포렌식](#)

[sbousseaden - Slides](#)