백신 일부 기능 제작

10101 강희찬 / 10306 박민기

제작 여정

05/17

세미나 주제 확정

세미나 주제를 정하려고 토의 하던중 최종적으로 백신프로그램 일부 기능을 제작 결정.

05/20

구현 방법 결정

API 사용과 바이러스 탐지 기능은 어떤식으로 구현할지 토의하고 최종 결정.

05/24

제작 시작

서로 본격적으로 제작을 시작한 날

06/15

제작 완료

코드를 합치고 ppt까지 전부 제작 완료.

06/17

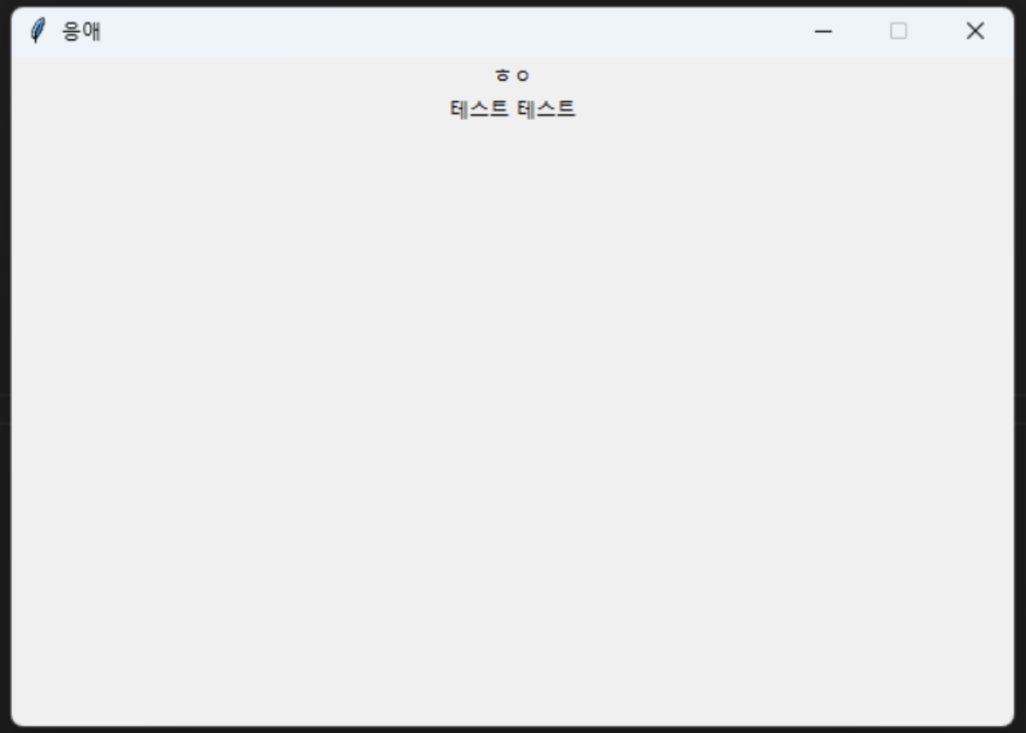
세미나 발표

원리 설명

GUI 초기 단계

C: > Users > user > Downloads > virus_test_1 > python_2 > import tkinter.py > ...

```python
import tkinter

window=tkinter.Tk()

window.title("응애")
window.geometry("600x400+700+320")
window.resizable(0,0)

label=tkinter.Label(window, text="ㅎㅇ")
label.pack()

HI = tkinter.Label(window, text="테스트 테스트")
HI.pack()

window.mainloop()
```



Tkinter

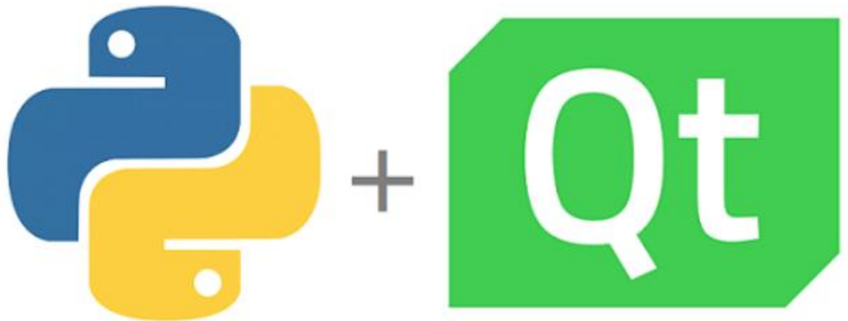
TKINTER

Tkinter는 버튼 등의 그래픽 기능을 제공해주는
그래픽 유저 인터페이스 모듈

단점

GUI디자인이 힘들다

```
btn = Button(root)                # root라는 창에 버튼 생성
btn.config(text=count)            # 버튼 내용
btn.config(width=10)              # 버튼 크기
btn.config(command=btnpress) # 버튼 기능 (btnpress() 함수 호출)
btn.pack()                        # 버튼 배치
```


Qt

Qt Designer

파일(F) 편집(E) 폼(O) 보기(V) 설정(S) 창(W) 도움말(H)

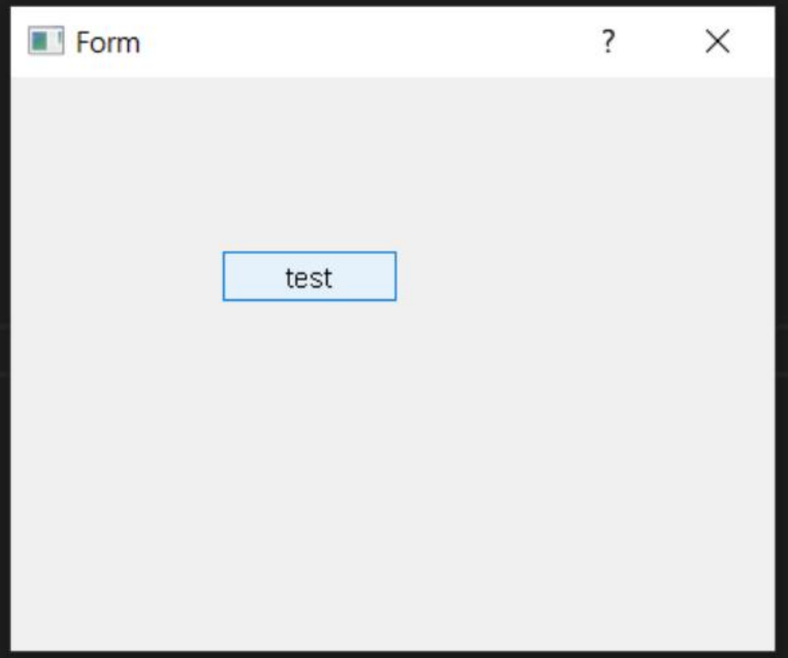
10306 박민기

project > python ㅇㅇ > dd.py > WindowClass > btnClick

위젯 상자

필터

```python
import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication, QDialog

form_class = uic.loadUiType("C:/project/python ㅇㅇ/Test_UI.ui")[0]

class WindowClass(QDialog, form_class):
    def __init__(self):
        super().__init__()
        self.setupUi(self)

        self.testbut.clicked.connect(self.btnClick)

    def btnClick(self) :
        print("어ㅏ나고ㅏㅣ로ㅓㅣ안몰ㅇㄴ멀")

if __name__ == "__main__":
    app = QApplication(sys.argv)
    myWindow = WindowClass()
    myWindow.show()
    sys.exit(app.exec_())
```



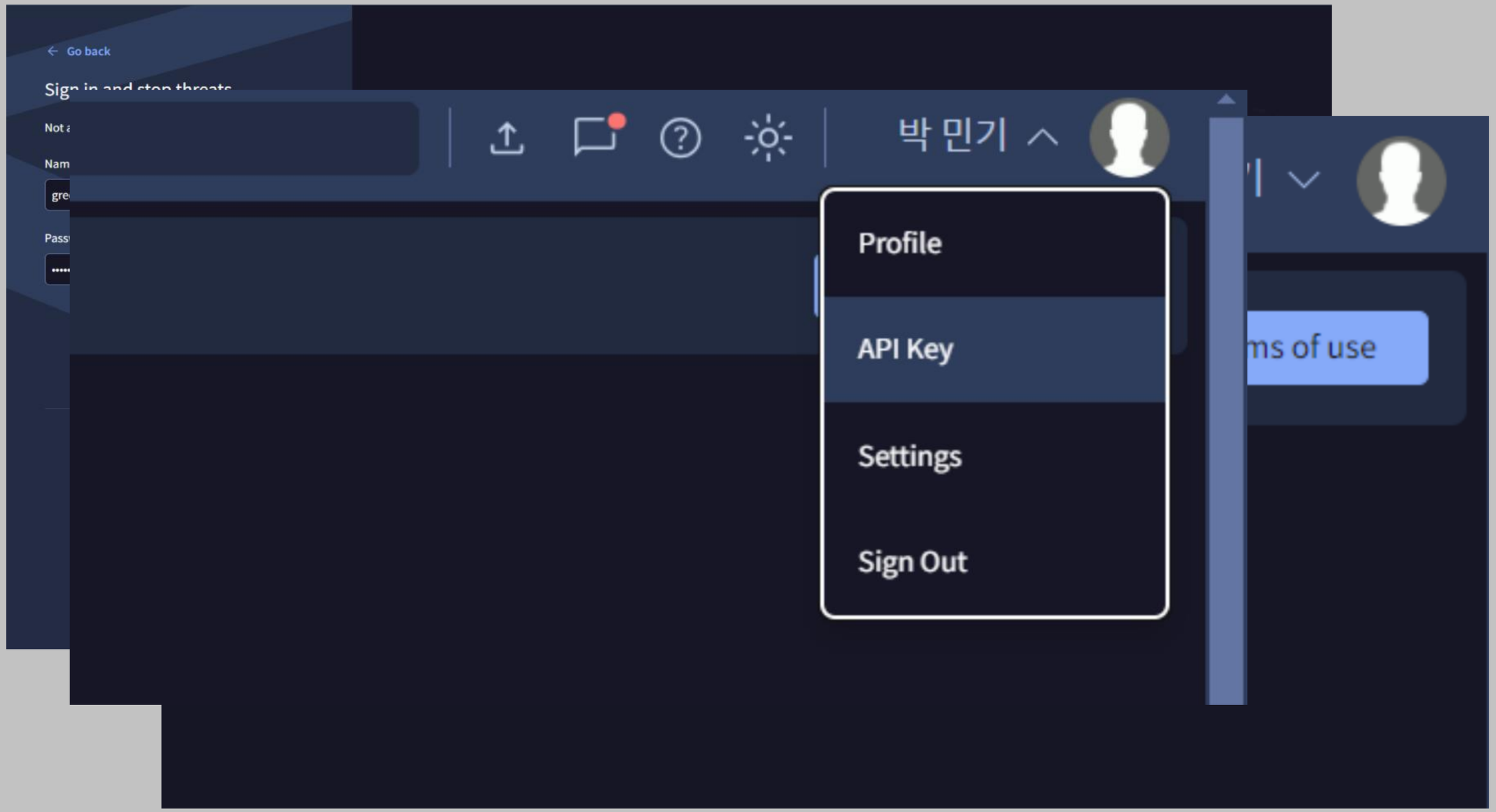

VIRUSTOTAL

API란 ?

Application Programming Interface(애플리케이션 프로그램 인터페이스)

다른 소프트웨어의 XX기능을 쓰고싶다..

[다른 소프트웨어]
특징 : XX기능 소유


Go back
박 민기
Profile
API Key
Settings
Sign Out
ms of use

vt-py

문서 검색

내용물:

설치하는 방법

개요

빠른 시작

API 참조

설치하는 방법

*vt-py*를 설치하는 가장 쉽고 권장되는 방법은 *pip*를 사용하는 것입니다 .

```
$ pip install vt-py
```

또는 GitHub에서 직접 소스 코드를 가져와 *setup.py* 를 실행할 수 있습니다 . 코드를 얻으려면 공개 저장소를 복제할 수 있습니다.

```
$ git clone git://github.com/VirusTotal/vt-py.git
$ cd vt-py
```

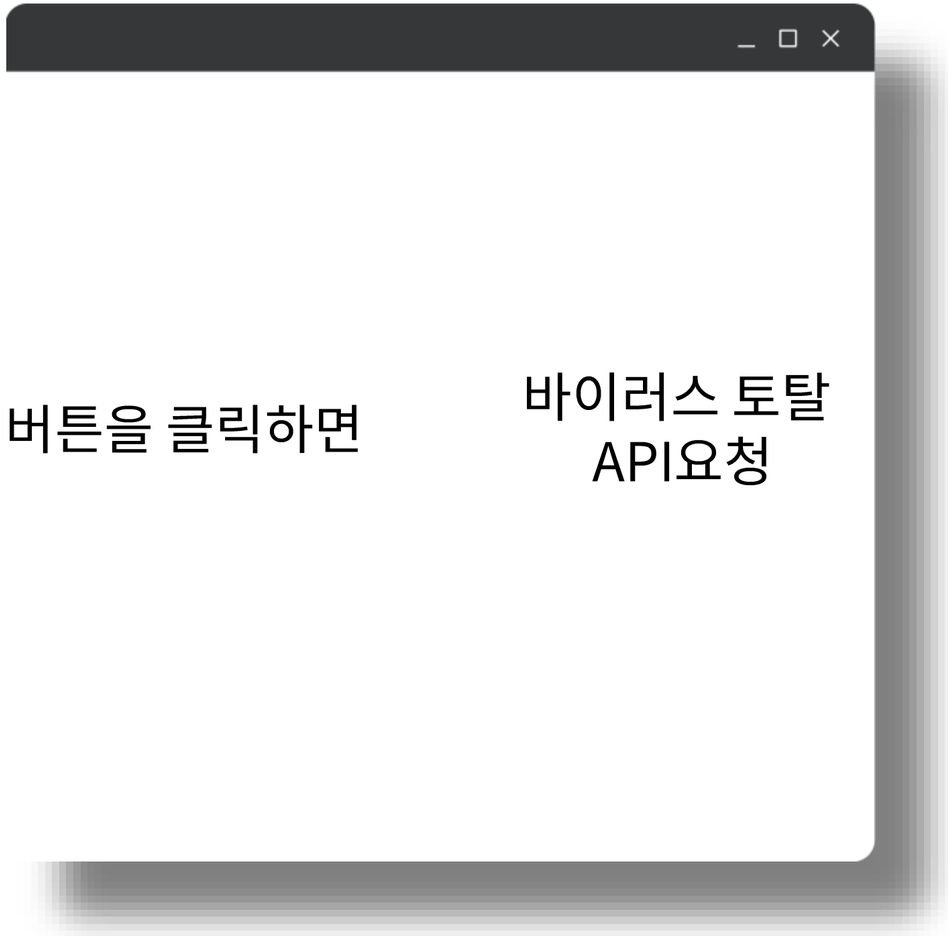
또는 최신 릴리스 용 tarball을 다운로드 하고 압축을 해제합니다.

```
$ tar -zxvf vt-py-X.Y.Z.tar.gz
$ cd vt-py-X.Y.Z
```

코드가 있으면 다음을 사용하여 설치할 수 있습니다.

```
$ sudo python3 setup.py install
```

이전의 다음


버튼을 클릭하면
바이러스 토탈
API요청

```python
import requests

url = "https://www.virust

headers = {"accept": "app

response = requests.get(u

print(response.text)
```

Virustotal API v3

Virustotal API

```
1  import virustotal_python
```

{'md5': '222e948e9b0a9addf43963021179772e', 'permalink': 'https://www.virustotal.com/gui/file/bfcaf79a9e814fd346b5842e6b3ce52cb784a1bafb931d8324582aaf639708f1/detection/f-bfcaf79a9e814fd346b5842e6b3ce52cb784a1bafb931d8324582aaf639708f1-1716595021', 'positives': 0, 'resource': '222e948e9b0a9addf43963021179772e', 'response_code': 1, 'scan_date': '2024-05-24 23:57:01', 'scan_id': 'bfcaf79a9e814fd346b5842e6b3ce52cb784a1bafb931d8324582aaf639708f1-1716595021', 'scans': {'ALYac': {'detected': False, 'result': None, 'update': '20240524', 'version': '2.0.0.10'}, 'AVG': {'detected': False, 'result': None, 'update': '20240524', 'version': '23.9.8494.0'}, 'Acronis': {'detected': False, 'result': None, 'update': '20240328', 'version': '1.2.0.121'}, 'AhnLab-V3': {'detected': False, 'result': None, 'update': '20240524', 'version': '3.25.1.10473'}, 'Antiy-AVL': {'detected': False, 'result': None, 'update': '20240524', 'version': '3.0'}, 'Arcabit': {'detected': False, 'result': None, 'update': '20240524', 'version': '2022.0.0.18'}, 'Avast': {'detected': False, 'result': None, 'update': '20240524', 'version': '23.9.8494.0'}, 'Avira': {'detected': False, 'result': None, 'update': '20240524', 'version': '8.3.3.18'}, 'Baidu': {'detected': False, 'result': None, 'update': '20190318', 'version': '1.0.0.2'}, 'BitDefender': {'detected': False, 'result': None, 'update': '20240524', 'version': '7.2'}, 'BitDefenderTheta': {'detected': False, 'result': None, 'update': '20240422', 'version': '7.2.37796.0'}, 'Bkav': {'detected': False, 'result': None, 'update': '20240524', 'version': '2.0.0.1'}, 'CAT-QuickHeal': {'detected': False, 'result': None, 'update': '20240524', 'version': '22.00'}, 'CMC': {'detected': False, 'result': None, 'update': '20240524', 'version': '2.4.2022.1'}, 'ClamAV': {'detected': False, 'result': None, 'update': '20240524', 'version': '1.3.1.0'}, 'CrowdStrike': {'detected': False, 'result': None, 'update': '20231026', 'version': '1.0'}, 'Cybereason': {'detected': False, 'result': None, 'update': '20240512', 'version': '1.2.449'}, 'Cynet': {'detected': False, 'result': None, 'update': '20240524', 'version': '4.0.1.1'}, 'DrWeb': {'detected': False, 'result': None, 'update': '20240524', 'version': '7.0.65.5230'}, 'ESET-NOD32': {'detected': False, 'result': None, 'update': '20240524', 'version': '29282'}, 'Emsisoft': {'detected': False, 'result': None, 'update': '20240524', 'version': '2024.1.0.53752'}, 'F-Secure': {'detected': False, 'result': None, 'update': '20240524', 'version': '18.10.1547.307'}, 'FireEye': {'detected': False, 'result': None, 'update': '20240524', 'version': '35.47.0.0'}, 'Fortinet': {'detected': False, 'result': None, 'update': '20240524', 'version': 'None'}, 'GData': {'detected': False, 'result': None, 'update': '20240524', 'version': 'A:25.38057B:27.36116'}, 'Google': {'detected': False, 'result': None, 'update': '20240524', 'version': '1716589845'}, 'Gridinsoft': {'detected': False, 'result': None, 'update': '20240524', 'version': '1.0.177.174'}, 'Ikarus': {'detected': False, 'result': None, 'update': '20240524', 'version': '6.3.12.0'}, 'Jiangmin': {'detected': False, 'result': None, 'update': '20240523', 'version': '16.0.100'}, 'K7AntiVirus': {'detected': False, 'result': None, 'update': '20240524', 'version': '12.160.52088'}, 'K7GW': {'detected': False, 'result': None, 'update': '20240524', 'version': '12.160.52089'}, 'Kaspersky': {'detected': False, 'result': None, 'update': '20240524', 'version': '22.0.1.28'}, 'Kingsoft': {'detected': False, 'result': None, 'update': '20230906', 'version': 'None'}, 'Lionic': {'detected': False, 'result': None, 'update': '20240524', 'version': '7.5'}, 'MAX': {'detected': False, 'result': None, 'update': '20240524', 'version': '2023.1.4.1'}, 'Malwarebytes': {'detected': False, 'result': None, 'update': '20240524', 'version': '4.5.5.54'}, 'MaxSecure': {'detected': False, 'result': None, 'update': '20240524', 'version': '1.0.0.1'}, 'McAfee': {'detected': False, 'result': None, 'update': '20240524', 'version': '6.0.6.653'}, 'MicroWorld-eScan': {'detected': False, 'result': None, 'update': '20240524', 'version': '14.0.409.0'}, 'Microsoft': {'detected': False, 'result': None, 'update': '20240524', 'version': '1.1.24040.1'}, 'NANO-Antivirus': {'detected': False, 'result': None, 'update': '20240524', 'version': '1.0.146.25796'}, 'Panda': {'detected': False, 'result': None, 'update': '20240524', 'version': '4.6.4.2'}, 'Rising': {'detected': False, 'result': None, 'update': '20240524', 'version': '25.0.0.27'}, 'SUPERAntiSpyware': {'detected': False, 'result': None, 'update': '20240524', 'version': '5.6.0.1032'}, 'Sangfor': {'detected': False, 'result': None, 'update': '20240521', 'version': '2.23.0.0'}, 'Skyhigh': {'detected': False, 'result': None, 'update': '20240524', 'version': 'v2021.2.0+4045'}, 'Sophos': {'detected': False, 'result': None, 'update': '20240524', 'version': '2.5.5.0'}, 'Symantec': {'detected': False, 'result': None, 'update': '20240524', 'version': '1.21.0.0'}, 'TACHYON': {'detected': False, 'result': None, 'update': '20240524', 'version': '2024-05-24.02'}, 'Tencent': {'detected': False, 'result': None, 'update': '20240524', 'version': '1.0.0.1'}, 'TrendMicro': {'detected': False, 'result': None, 'update': '20240524', 'version': '11.0.0.1006'}, 'TrendMicro-HouseCall': {'detected': False, 'result': None, 'update': '20240524', 'version': '10.0.0.1040'}, 'VBA32': {'detected': False, 'result': None, 'update': '20240524', 'version': '5.0.0'}, 'VIPRE': {'detected': False, 'result': None, 'update': '20240524', 'version': '6.0.0.35'}, 'Varist': {'detected': False, 'result': None, 'update': '20240524', 'version': '6.5.1.2'}, 'ViRobot': {'detected': False, 'result': None, 'update': '20240524', 'version': '2014.3.20.0'}, 'VirIT': {'detected': False, 'result': None, 'update': '20240524', 'version': '9.5.709'}, 'Xcitium': {'detected': False, 'result': None, 'update': '20240524', 'version': '36729'}, 'Yandex': {'detected': False, 'result': None, 'update': '20240524', 'version': '5.5.2.24'}, 'Zillya': {'detected': False, 'result': None, 'update': '20240524', 'version': '2.0.0.5120'}, 'ZoneAlarm'2.0'}, 'alibabacloud': {'detected': False, 'result': None, 'update': '20240513', 'version': '2.1.0'}, 'tehtris': {'detected': False, 'result': None, 'update': '20240524', 'version': 'v0.1.4'}}, 'sha1': '0f2ce4696f7a0e7b7f2817b93713b14564034613', 'sha256': 'bfcaf79a9e814fd346b5842e6b3ce52cb784a1bafb931d8324582aaf639708f1', 'total': 64, 'verbose_msg': 'Scan finished, information embedded'}

```
17      ##print(response.json())
18
```

```
16      response = requests.get(url, params=params)
17      ##print(response.json())
18
19      report = response.json()
20      report_scan_date = report.get('scan_date')
21      report_scan_sha256 = report.get('sha256')
22      report_scan_md5 = report.get('md5')
23      report_scan_result = report.get('scans')
24      report_scan_vendors = list(report['scans'].keys())
25      report_scan_vendors_cnt = len(report_scan_vendors)
26
27
28      print('Scan Date (UTC) : ', report_scan_date)
29      print('Scan File SHA256 : ', report_scan_sha256)
30      print('Scan File MD5 : ', report_scan_md5)
31      print('Scan File Vendor CNT : ', report_scan_vendors_cnt, '\n')
32
```

Json 형식에서 'md5'value값 가져와
report_scan_md5에 저장

report_scan_md5(저장된 변수(Value값))을 프린트 함

비슷한 방식으로 vendor별로 for문을 돌면서 정보출력

```
37      for vendor in report_scan_vendors:
38          outputs = report_scan_result[vendor]
39          outputs_result = report_scan_result[vendor].get('result')
40          outputs_version = report_scan_result[vendor].get('version')
41          outputs_detected = report_scan_result[vendor].get('detected')
42          outputs_update = report_scan_result[vendor].get('update')
43
44          print('No', num,
45            'Vendor Name :', vendor,
46            ', Vendor Version :', outputs_version,
47            ', Scan Detected :', outputs_detected,
48            ', Scan Result :', outputs_result)
49          num = num + 1
```

Scan Date (UTC) : 2024-05-24 23:57:01

Scan File SHA256 : bfcaf79a9e814fd346b5842e6b3ce52cb784a1bafb931d8324582aaf639708f1
Scan File MD5 : 222e948e9b0a9addf43963021179772e
Scan File Vendor CNT : 64

보기 간단하게 출력

No 1 Vendor Name : ALYac , Vendor Version : 2.0.0.10 , Scan Detected : False , Scan Result : None
No 2 Vendor Name : AVG , Vendor Version : 23.9.8494.0 , Scan Detected : False , Scan Result : None
No 3 Vendor Name : Acronis , Vendor Version : 1.2.0.121 , Scan Detected : False , Scan Result : None
No 4 Vendor Name : AhnLab-V3 , Vendor Version : 3.25.1.10473 , Scan Detected : False , Scan Result : None
No 5 Vendor Name : Antiy-AVL , Vendor Version : 3.0 , Scan Detected : False , Scan Result : None
No 6 Vendor Name : Arcabit , Vendor Version : 2022.0.0.18 , Scan Detected : False , Scan Result : None
No 7 Vendor Name : Avast , Vendor Version : 23.9.8494.0 , Scan Detected : False , Scan Result : None
No 8 Vendor Name : Avira , Vendor Version : 8.3.3.18 , Scan Detected : False , Scan Result : None
No 9 Vendor Name : Baidu , Vendor Version : 1.0.0.2 , Scan Detected : False , Scan Result : None
No 10 Vendor Name : BitDefender , Vendor Version : 7.2 , Scan Detected : False , Scan Result : None
No 11 Vendor Name : BitDefenderTheta , Vendor Version : 7.2.37796.0 , Scan Detected : False , Scan Result : None
No 12 Vendor Name : Bkav , Vendor Version : 2.0.0.1 , Scan Detected : False , Scan Result : None
No 13 Vendor Name : CAT-QuickHeal , Vendor Version : 22.00 , Scan Detected : False , Scan Result : None
No 14 Vendor Name : CMC , Vendor Version : 2.4.2022.1 , Scan Detected : False , Scan Result : None
No 15 Vendor Name : ClamAV , Vendor Version : 1.3.1.0 , Scan Detected : False , Scan Result : None
.
.
.
.
.
.

새 창 띄우기

GUI 창 여러개 띄우는법

```
class first(QDialog):
    def __init__(self):
```

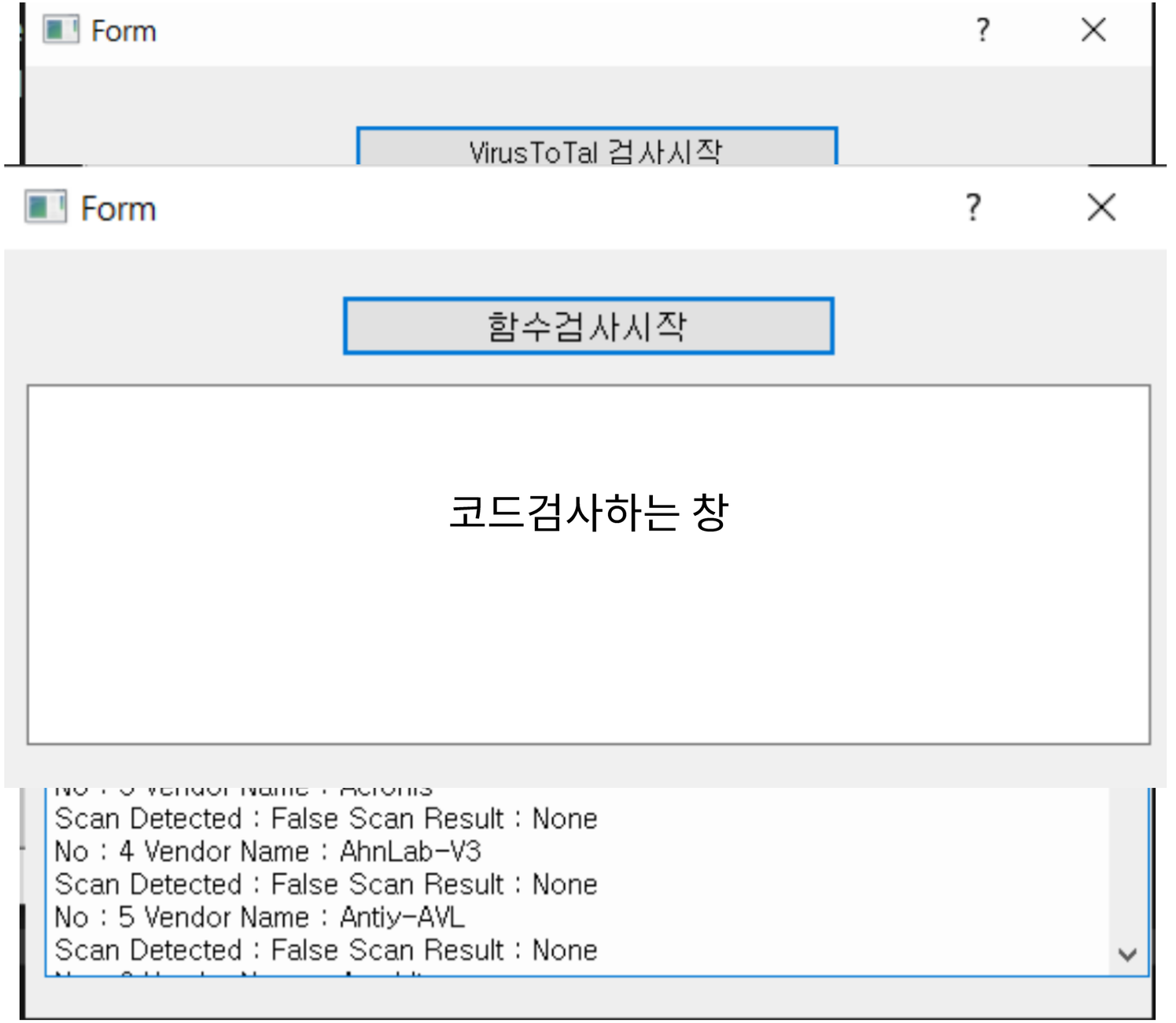
메인화면 띄우는 클래스

```
class second(QDialog):
    def __init__(self):
        super().__init__()
        self.ui = uic.loadUi("multi_2.ui", self)
        self.show()
```

두번째(새로운) 창을 띄움

```
    def second_window(self):
        window_2 = second()
        window_2.exec_()
```

window_2클래스도 연결


Form
VirusToTal 검사시작
Form
함수검사시작
코드검사하는 창
Scan Detected : False Scan Result : None
No : 4 Vendor Name : AhnLab-V3
Scan Detected : False Scan Result : None
No : 5 Vendor Name : Antiy-AVL
Scan Detected : False Scan Result : None

test 9+ dd.py 1

project > python_2 > test > ...

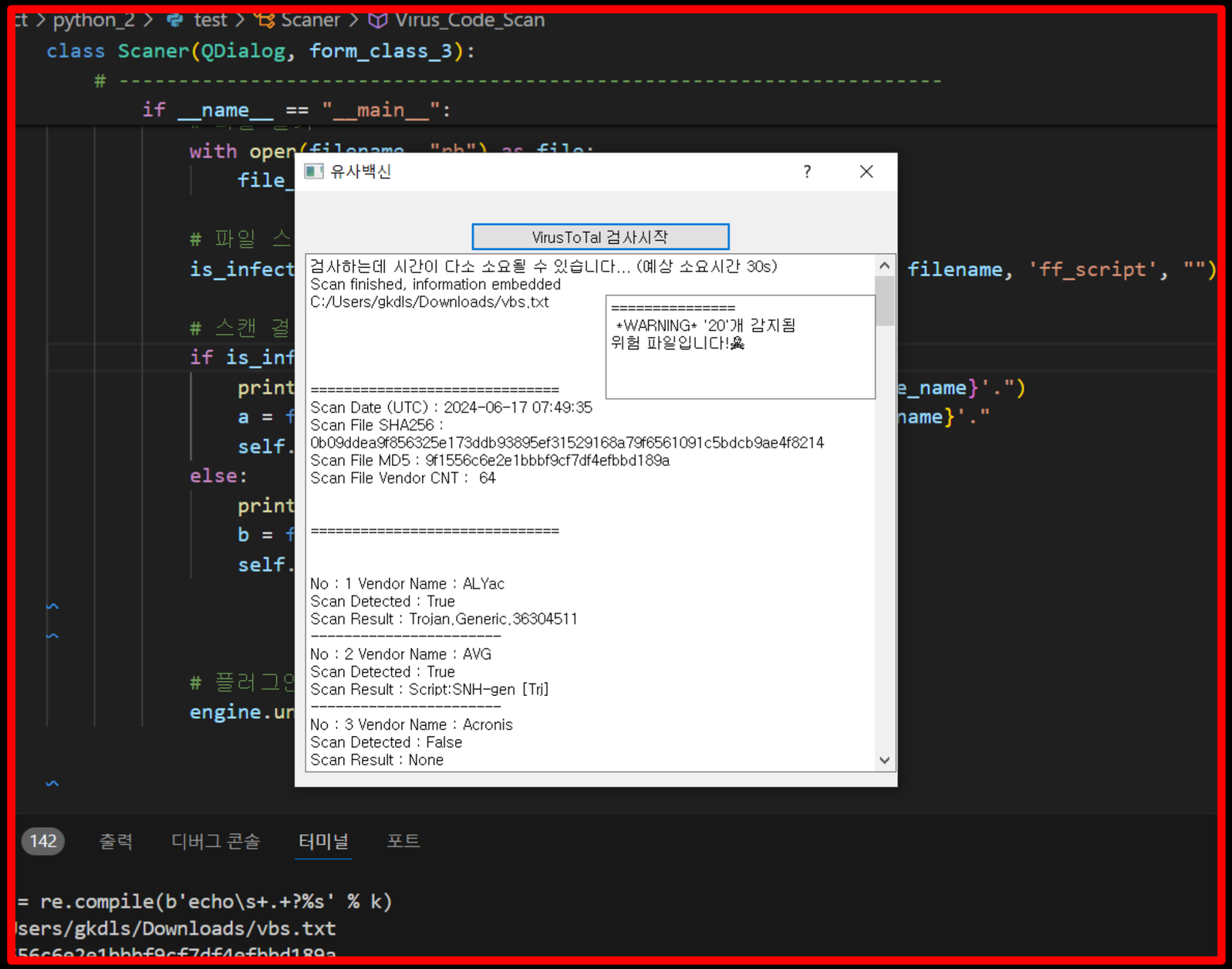
```python
import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication, QDialog, QFileDialog
import virustotal_python, json, requests
import requests, time
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
import re
import os
import hashlib
import kernel
import kavutil

form_class = uic.loadUiType("C:\project\python_2\GUI\PyQt_GUI_Designer.ui")[0]
form_class_2 = uic.loadUiType("C:\project\python_2\GUI\API_GUI.ui")[0]
form_class_3 = uic.loadUiType("C:\project\python_2\GUI\Scaner_GUI.ui")[0]

class WindowClass(QDialog, form_class):
    def __init__(self):
        super().__init__()
        self.setupUi(self)

        self.print.clicked.connect(self.btnClick)
        self.dialog.clicked.connect(self.open_file_dialog)
```

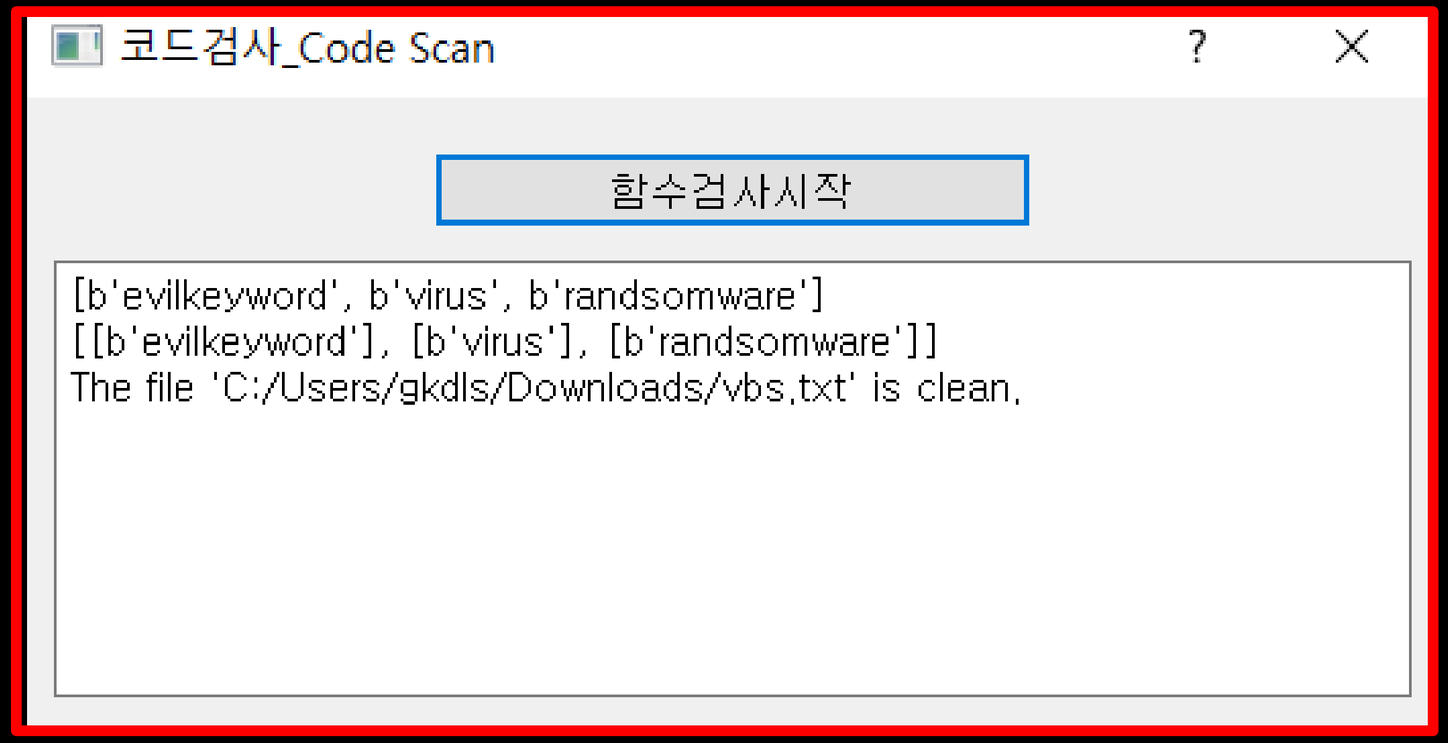
문제 130 출력 디버그 콘솔 터미널 포트

Python
Python

```
C:/project/python_2/test
954de761ba0eb6e3d7282199a480d12c
Scan finished, information embedded

[b'evilkeyword', b'virus', b'randsomware']
[[b'evilkeyword'], [b'virus'], [b'randsomware']]
The file 'C:/project/python_2/test' is infected with 'Virus.Pattern'.
```

줄 13, 열 1 공백: 4 UTF-8 CRLF Python 3.12.2 64-bit


t > python_2 > test > Scaner > Virus_Code_Scan
class Scaner(QDialog, form_class_3):
# ----------
if __name__ == "__main__":
with open(filename, "rb") as file:
file_
파일 스
is_infect
filename, 'ff_script', "")
스캔 결
if is_inf
print
e_name}'.")
a = f
name}'."
self.
else:
print
b = f
self.
플러그인
engine.ur
142
출력
디버그 콘솔
터미널
포트
= re.compile(b'echo\s+.+?%s' % k)
Users/gkdls/Downloads/vbs.txt
유사백신
VirusToTal 검사시작
검사하는데 시간이 다소 소요될 수 있습니다... (예상 소요시간 30s)
Scan finished, information embedded
C:/Users/gkdls/Downloads/vbs.txt
==========
WARNING '20'개 감지됨
위험 파일입니다!
==========
Scan Date (UTC) : 2024-06-17 07:49:35
Scan File SHA256 :
0b09ddea9f856325e173ddb93895ef31529168a79f6561091c5bdcb9ae4f8214
Scan File MD5 : 9f1556c6e2e1bbbf9cf7df4efbbd189a
Scan File Vendor CNT : 64
==========
No : 1 Vendor Name : ALYac
Scan Detected : True
Scan Result : Trojan.Generic.36304511
No : 2 Vendor Name : AVG
Scan Detected : True
Scan Result : Script:SNH-gen [Trj]
No : 3 Vendor Name : Acronis
Scan Detected : False
Scan Result : None


코드검사_Code Scan
?
함수검사시작
[b'evilkeyword', b'virus', b'randsomware']
[[b'evilkeyword'], [b'virus'], [b'randsomware']]
The file 'C:/Users/gkdls/Downloads/vbs.txt' is clean.

SCAN

주석 제거

```
1  def init(self, plugins_path, verbose=False):
2      self.p_js_comments = re.compile(rb'(//.*|/\*.*?\*/)', re.IGNORECASE | re.DOTAL
3  L)  self.p_vbs_comments = re.compile(rb'\'[^\n]*', re.IGNORECASE)
4      self.p_remove_spaces = re.compile(rb'[\s]+')
5      return 0
6
7  def uninit(self):
8      return 0
```

기본적으로 거의 바이러스 파일들은 난독화가 되어있어서 보기가 힘듭니다.

주석을 이용하여서 시그니처 기반탐색을 회피하기도 해서 주석을 제거시켜줍니다.
그리고 uninit을 이용하여서 리소스를 정리와 동시에 코드를 종료시켜 줍니다.

```python
def scan(self, filecontent, filename,id,infect):
    try:
        mm = filecontent
        if filename.endswith(('.exe', '.js', '.dll','.bat')):
            buf = mm[:]
            buf = self.p_js_comments.sub(b'', buf)
            buf = self.p_remove_spaces.sub(b'', buf)
            buf = buf.lower()
            for pattern in vdb:
                if pattern[0] in buf:
                    return True, 'Virus detection', 0, INFECTED
    except IOError:
        pass
    except ValueError:
        pass
    return False, '', -1, NOT_FOUND
```

파일 확장자가
.exe,.js,.dll,.bat
와 같은형식 이라면 주석 제거 후
패턴과 비교

loadvirusDB 와 makevirusDB
를 통하여서 .DB파일로 저장되어 있는
바이러스가 자주쓰는
모듈리스트를 불러와 리스트에
:을 기준으로 나눠서 저장시킵니다.

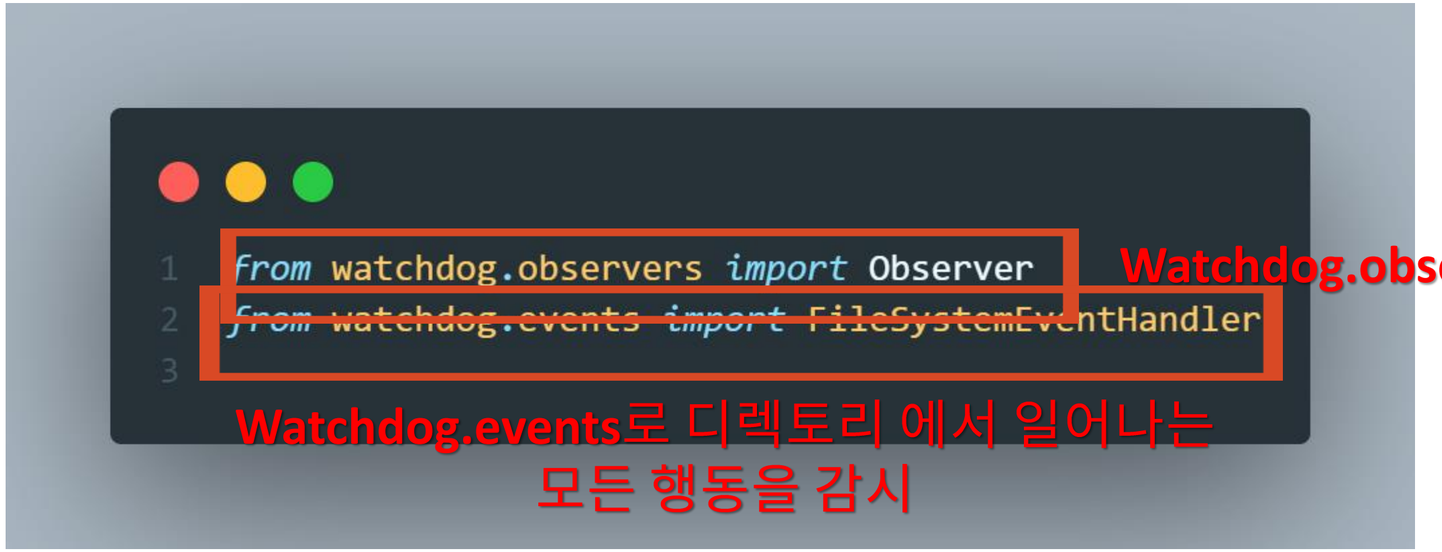
```
1  def loadvirusDB(self):
2      with open("byeonsu.db", 'rb') as f
3  p:      for line in fp:
4              VirusDB.append(line.strip())
5
6  def makevirusDB(self):
7      for i in VirusDB:
8          vdb.append(i.split(b':'))
```

```python
def disinfect(self, filename, malware_id):
    try:
        if malware_id == 0:
            os.remove(filename)
            return True
    except IOError:
        pass

    return False

```

위에 SCAN 함수 부분에서 받은 malware_id를 통하여서 치료 방법을 결정해줍니다.

실시간 탐지

실시간 탐지


1 from watchdog.observers import Observer
2 from watchdog.events import FileSystemEventHandler
3
Watchdog.observer로 디렉토리 감지
Watchdog.events로 디렉토리 에서 일어나는
모든 행동을 감시

실시간 탐지


def load_signatures(file_path, buffer_size=1024):
signatures = {}
while True:
data = f.read(buffer_size)
if not data:
break
for line in lines:
stripped_line = line.strip()
if stripped_line:
signatures[stripped_line] = stripped_line
return signatures
1024바이트 까지 읽고
줄 단위로 분리 반복
공백 제거 후 signature안에
저장

실시간 탐지

파일의 끝까지 4096바이트씩 읽어서 MD5 객체 업데이트

```python
def file_md5(file_path):
    hash_md5 = hashlib.md5()
    with open(file_path, "rb") as f:
        for chunk in iter(lambda: f.read(4096), b""):
            hash_md5.update(chunk)
    return hash_md5.hexdigest()
```

실시간 탐지

Content 파일에서 리스트 내의 문자열이 하나라도 있다면 바이러스 취급

```python
def is_malicious(self, file_path):
    suspicious_imports = ['바이러스들이 많이호출하는 모듈']

    try:
        with open(file_path, 'rb') as f:
            content = f.read()
            if any(import_str in content for import_str in suspicious_imports):
                return True
    except FileNotFoundError:
        print("파일을 찾을 수 없습니다.")
        return False

    return False
```

실시간 탐지

파일해시가 시그니처 파일 안에 있다면 삭제
파일 내용에 바이러스가 많이 호출하는 모듈 리스트안에 있다면 격리

```python
if file_hash in self.virus_signatures:
    print(f"Virus detected: {self.virus_signatures[file_hash]} in {file_path}")
    self.take_action(file_path, 'delete')
    return
if self.is_malicious(file_path):
    print(f"Potentially malicious behavior detected in {file_path}")
    self.take_action(file_path, 'quarantine')
    return
print(f"No virus detected in {file_path}")
```

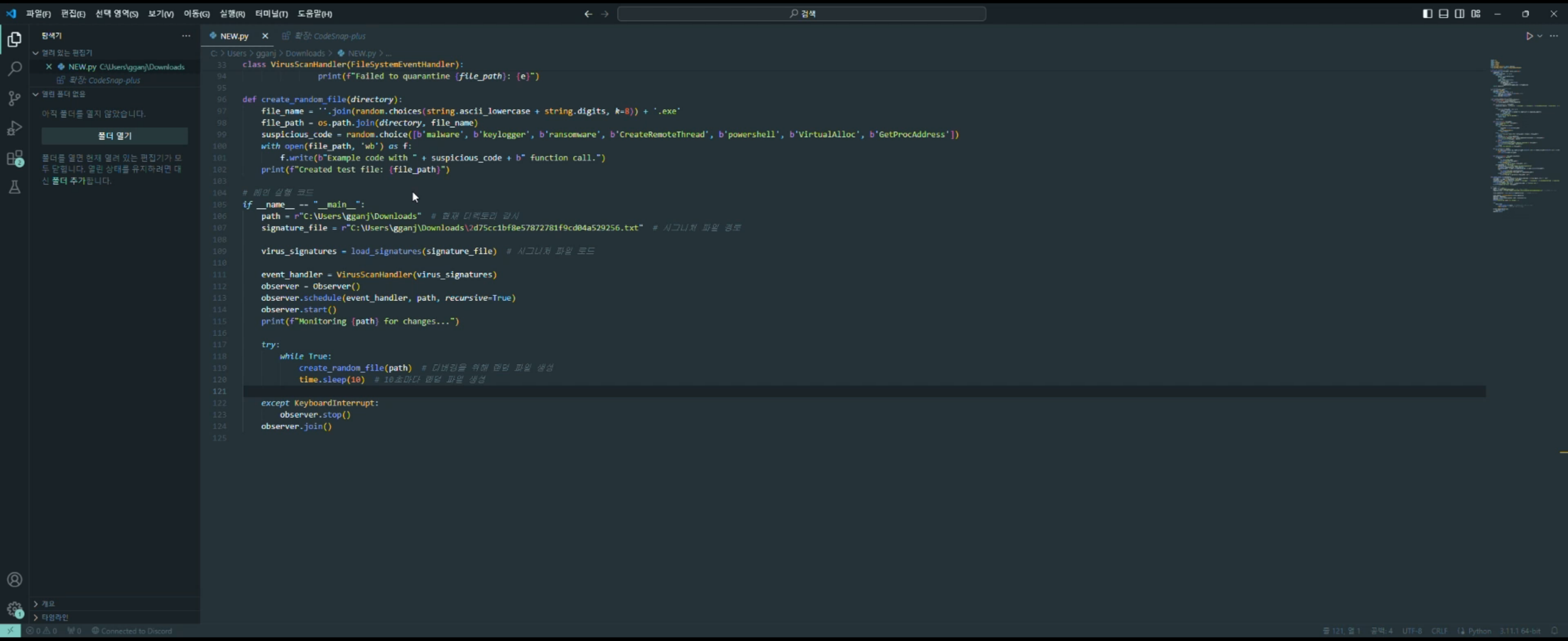
실시간 탐지


```
1  def on_created(self, event):
2      if not event.is_directory and (event.src_path.endswith('.exe') or event.src_path.endswith('.dll') or event.src_path.endswith('.ba
3  t')):   time.sleep(1)
4          self.scan_file(event.src_path)
```

디렉토리 이벤트가 아닌 파일 이벤트고
특정 확장자라면 스캔

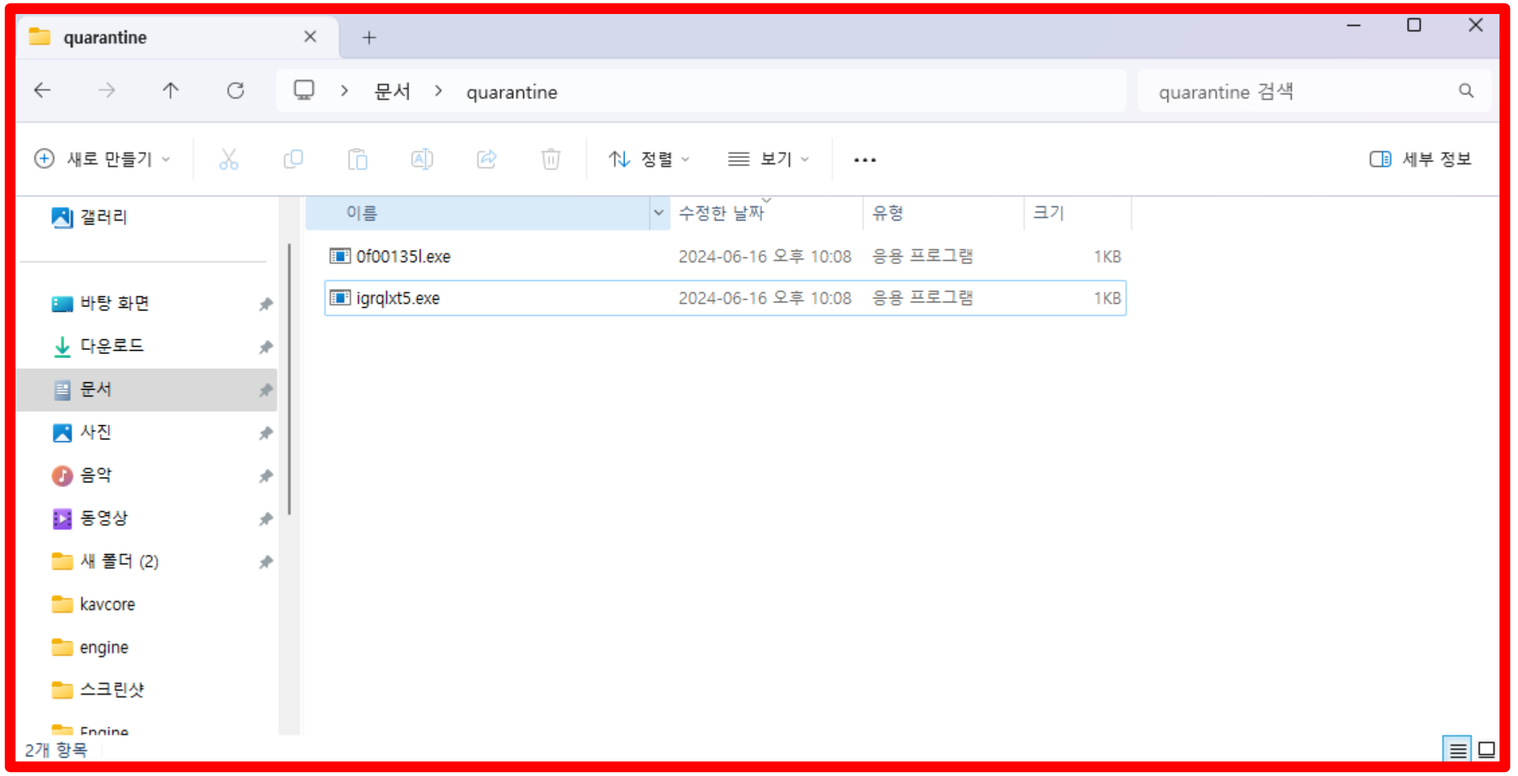
실시간 탐지 치료 방법이 delete라면 삭제 quarantine 이라면 격리

```python
def take_action(self, file_path, disinfect):
    if disinfect == 'delete':
        try:
            os.remove(file_path)
            print(f"Deleted {file_path}")
        except Exception as e:
            print(f"Failed to delete {file_path}: {e}")
    elif disinfect == 'quarantine':
        quarantine_dir = r"C:\Users\gganj\Documents\quarantine"
        os.makedirs(quarantine_dir, exist_ok=True)
        quarantine_path = os.path.join(quarantine_dir, os.path.basename(file_path))
        try:
            os.rename(file_path, quarantine_path)
            print(f"Quarantined {file_path} to {quarantine_path}")
        except Exception as e:
            print(f"Failed to quarantine {file_path}: {e}")
```


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)
검색
탐색기
열려 있는 편집기
NEW.py C:\Users\gganj\Downloads
확장: CodeSnap-plus
열린 폴더 없음
아직 폴더를 열지 않았습니다.
폴더 열기
폴더를 열면 현재 열려 있는 편집기가 모두 닫힙니다. 열린 상태를 유지하려면 대신 폴더 추가합니다.
개요
타임라인
NEW.py
확장: CodeSnap-plus
C: > Users > gganj > Downloads > NEW.py > ...
class VirusScanHandler(FileSystemEventHandler):
print(f"Failed to quarantine {file_path}: {e}")
def create_random_file(directory):
file_name = ''.join(random.choices(string.ascii_lowercase + string.digits, k=8)) + '.exe'
file_path = os.path.join(directory, file_name)
suspicious_code = random.choice([b'malware', b'keylogger', b'ransomware', b'CreateRemoteThread', b'powershell', b'VirtualAlloc', b'GetProcAddress'])
with open(file_path, 'wb') as f:
f.write(b"Example code with " + suspicious_code + b" function call.")
print(f"Created test file: {file_path}")
# 메인 실행 코드
if __name__ == "__main__":
path = r"C:\Users\gganj\Downloads" # 현재 디렉토리 감시
signature_file = r"C:\Users\gganj\Downloads\2d75cc1bf8e57872781f9cd04a529256.txt" # 시그니처 파일 경로
virus_signatures = load_signatures(signature_file) # 시그니처 파일 로드
event_handler = VirusScanHandler(virus_signatures)
observer = Observer()
observer.schedule(event_handler, path, recursive=True)
observer.start()
print(f"Monitoring {path} for changes...")
try:
while True:
create_random_file(path) # 디버깅을 위해 랜덤 파일 생성
time.sleep(10) # 10초마다 랜덤 파일 생성
except KeyboardInterrupt:
observer.stop()
observer.join()
Connected to Discord
줄 121, 열 1 공백: 4 UTF-8 CRLF Python 3.11.1 64-bit

실시간 탐지

실행이 잘된 모습



Q & A

END

10101 강희찬 / 10306 박민기