Anti-Virus Program
개발 발표

중부대학교 손경현 · 김채원

00

Table of Contents

CCIT
CULTURE CONTENTS & INFORMATION TECHNOLOGY

서론

- 팀원 소개
- 주제 소개
- 개발 동기 설명

Anti-Virus Program 소개

- 프로그램 및 주요 코드 설명
- 프로그램 실행 영상 재생
- 행위 기반을 통한
  랜섬웨어 탐지 기능 설명

향후 계획 및 마무리

- 미완성 및 부족한 부분
- 향후 진행 상황
- Q&A

팀원 소개

손경현(조장)

[프로그래밍, 디자인 설계]

김채원

[기능과 성능 테스트, 조사 등]

주제 소개

연합뉴스 구독

SNS 해킹 의뢰로 웹사이트 고객정보 수백만건 탈취

입력 2023.11.01. 오전 11:33 · 수정 2023.11.01. 오전 11:34 기사원문

정회성 기자

추천 댓글 가

전남경찰, 해커·알선업자·판매책 등 13명 송치

TV CHOSUN 구독 PICK

선관위, 北 '킴수키'에 직원 메일 해킹 확인…"해킹시 투개표 결과 조작 가능"

입력 2023.10.10. 오후 12:31 · 수정 2023.10.10. 오후 4:11 기사원문

이채림 기자

822 468 가

중앙선거관리위원회의 업무망·내부망 등이 보안에 취약하고, 해킹 당할 경우 투표자의 사전 투표 여부나 개표 결과 등 중요한 정보들이 조작될 수 있다는 사실이 10일 확인됐다.

국정원은 이날 지난 7월 17일부터 지난달 22일까지 12주간 선관위에 대해 한국인터넷진흥원(KISA)과 합동으로 보안점검한 결과를 발표했다.

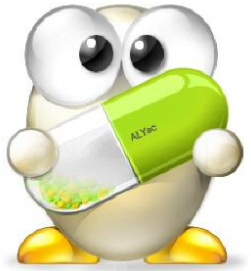
이번 보안 점검은 국정원과 선관위, KISA가 함께 진행했고, 여야 참관인이 입회한 가운데에 실시됐다.

백종욱 국정원 3차장은 "이번 점검은 기술적 측면에서 해커의 관점으로 시스템의 취약점을 확인한 것"이라며 "과거에 제기됐던 선거 관련 의혹들과 단순 결부 시키는 것은 경계해야한다"고 했다.

■ "선관위 해킹시 유권자의 투표 여부·개표 결과도 조작 가능"

늘어나는 **해킹 피해**, **안티 바이러스 프로그램** 중요성 부각 !

개발 동기 설명



다양한 Anti-Virus 프로그램이 존재하며 이에 다양한 Engine이 존재

→ Anti-Virus 프로그램을 제작함으로써 현존하는 바이러스의 검사 원리 이해 도모

02 - Anti-Virus Program 설명

02

Anti-Virus Program 설명

Program 설명

decoy.py

행위 기반 랜섬웨어 탐지

가짜 파일 생성

가짜 파일 감시 …

랜섬웨어가
가짜 파일 암호화

엔진에서 파일 변경 탐지

암호화한 주체를
랜섬웨어로 간주

Antivirus.py

악성코드의 해시값 등을 통한 판별

악성코드 의심 파일

VirusTotal

OPSWAT Metadefender

바이러스 여부 확인

(Virus Total, Meta Defeder의 Score 확인 가능 !)

Anti-Virus Program 설명

주요 code 설명


```
def removeFile(file):
        try:
            os.remove(file)
        except:
            # 파일을 삭제할 수 없으면 오류 보여주기
            msgBox = QtWidgets.QMessageBox()
            msgBox.setIcon(QtWidgets.QMessageBox.Critical)
            msgBox.setText("경고")
            msgBox.setInformativeText(f"""\
파일을 삭제할 수 없습니다.
파일: {file}"
            """)
            # 윈도우 타이틀 바 제거
            msgBox.setWindowFlags(QtCore.Qt.WindowStaysOnTopHint)
            msgBox.setWindowFlags(QtCore.Qt.FramelessWindowHint)
            msgBox.exec_()
        finally:
            # 파일이 제거되면 성공적으로 제거되었다는 알림
            msgBox = QtWidgets.QMessageBox()
            msgBox.setIcon(QtWidgets.QMessageBox.Information)
            msgBox.setText("안내")
            msgBox.setInformativeText(f"""\
파일이 성공적으로 제거되었습니다.
파일: {file}"
            """)
```

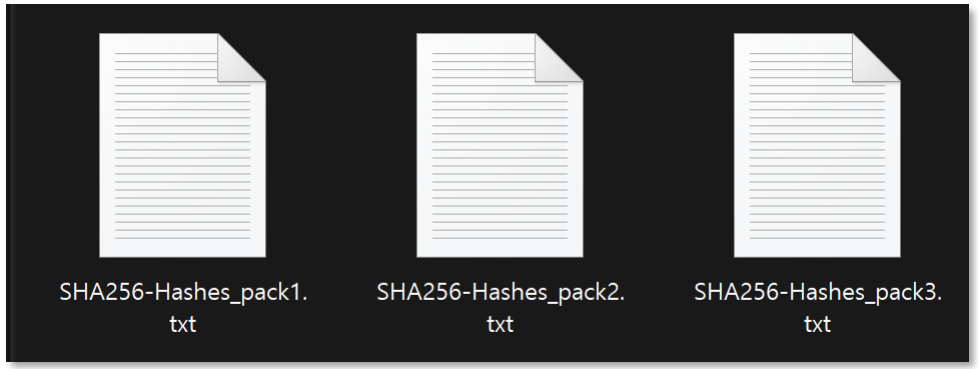
```
def displayResults_VIRUS(self, file):
    self.Tabs.setCurrentIndex(2)
    # 32MB 이하인지 검사
    if self.UseVirusTotalApiCheckBox.isChecked() and os.path.getsize(file) < 32000000:
        self.VirusTotalWidget.show()
    else:
        # hide Virus total results since it is not needed
        self.VirusTotalWidget.hide()
    # 메타 디펜더 120MB
    if self.UseMetaDefenderApiCheckBox.isChecked() and os.path.getsize(file) < 120000000:
        self.MetaDefenderWidget.show()
    else:
        # hide meta defender results since it is not needed
        self.MetaDefenderWidget.hide()
        self.IsFileVirusY_N.setStyleSheet("color: red")
        self.IsFileVirusY_N.setText("악성 파일 맞음")
    # 파일 삭제 버튼
    self.DeleteFileButton.clicked.connect(lambda: removeFile(file))
    # 돌아가기 버튼
    self.ReturnToHomeTabButton.clicked.connect(lambda: self.Tabs.setCurrentIndex(0))
```

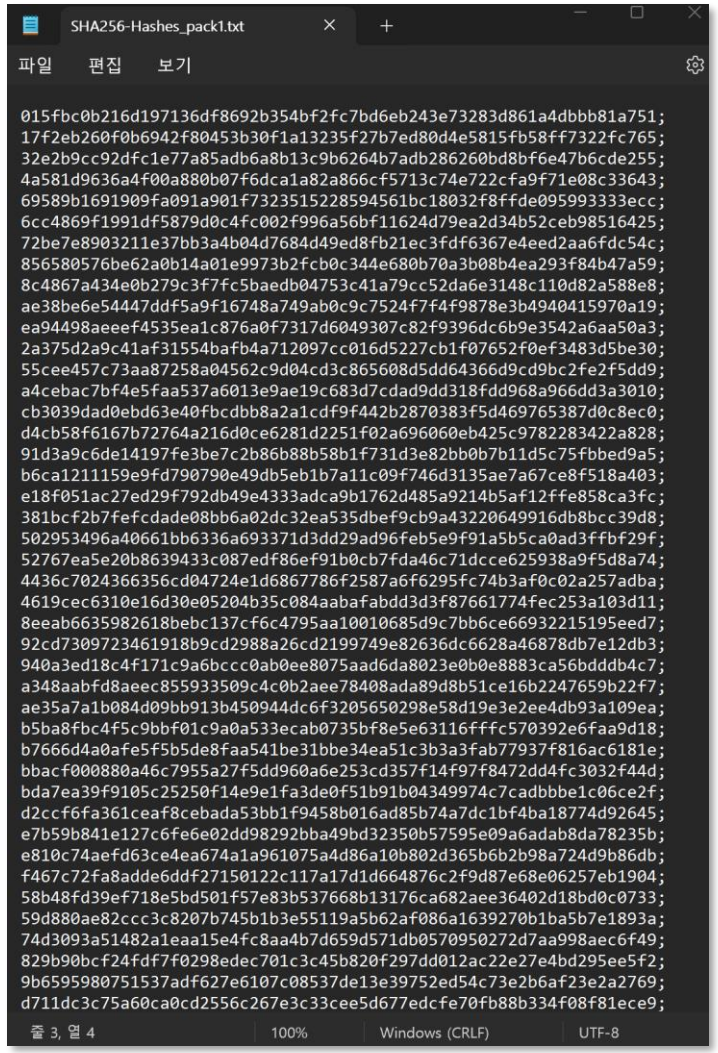
주요 code 설명


```
def scan(file, self, MainWindow):
    try:



        # 기본 값을 바이러스 찾는걸 false로 설정
        virus_found = False



        # 파일 열고 해시값 얻기
        with open(file,"rb") as f:
            bytes = f.read()
            readable_hash = hashlib.sha256(bytes).hexdigest();

        # 해시값 표시
        self.FileHash.setText("해시값:  " + readable_hash)

        # 나온 해시값 리스트에 있는지랑 비교하기

        # SHA256 HASHES check + pack 1
        with open(SHA256_HASHES_pack1,'r') as f:
            lines = [line.rstrip() for line in f]
            for line in lines:
                if str(readable_hash) == str(line.split(";")[0]):
                    virus_found = True
                    f.close()
        f.close()
```

```
finally:
    # goto hidden results tab
    self.Tabs.setCurrentIndex(2)

    # delete file button
    self.DeleteFileButton.clicked.connect(lambda: removeFile(file))
    # return button
    self.ReturnToHomeTabButton.clicked.connect(lambda: self.Tabs.setCurrentIndex(0))
```

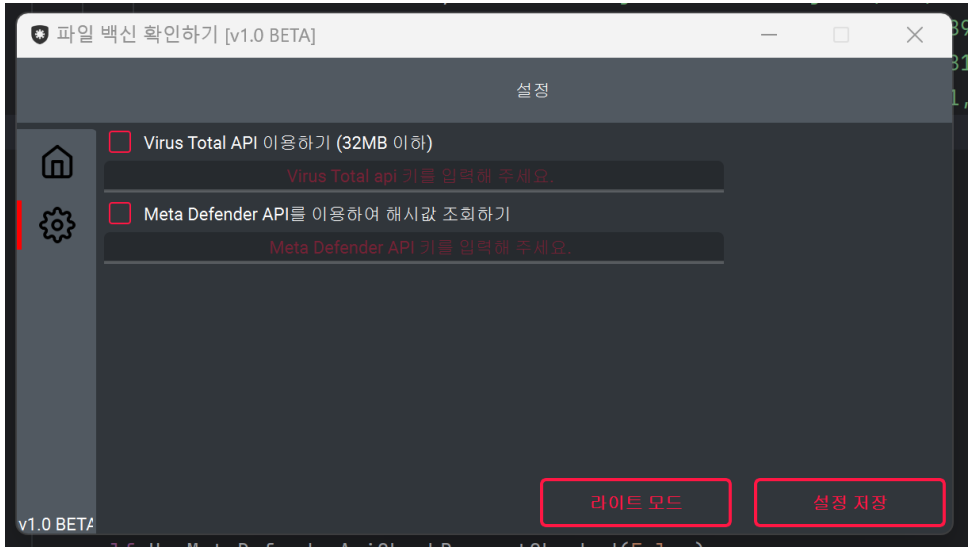
주요 코드 설명


SHA256-Hashes_pack1.
txt
SHA256-Hashes_pack2.
txt
SHA256-Hashes_pack3.
txt


SHA256-Hashes_pack1.txt
파일 편집 보기
015fbc0b216d197136df8692b354bf2fc7bd6eb243e73283d861a4dbbb81a751;
17f2eb260f0b6942f80453b30f1a13235f27b7ed80d4e5815fb58ff7322fc765;
32e2b9cc92dfc1e77a85adb6a8b13c9b6264b7adb286260bd8bf6e47b6cde255;
4a581d9636a4f00a880b07f6dca1a82a866cf5713c74e722cfa9f71e08c33643;
69589b1691909fa091a901f7323515228594561bc18032f8ffde095993333ecc;
6cc4869f1991df5879d0c4fc002f996a56bf11624d79ea2d34b52ceb98516425;
72be7e8903211e37bb3a4b04d7684d49ed8fb21ec3fdf6367e4eed2aa6fdc54c;
856580576be62a0b14a01e9973b2fcb0c344e680b70a3b08b4ea293f84b47a59;
8c4867a434e0b279c3f7fc5baedb04753c41a79cc52da6e3148c110d82a588e8;
ae38be6e54447ddf5a9f16748a749ab0c9c7524f7f4f9878e3b4940415970a19;
ea94498aeeef4535ea1c876a0f7317d6049307c82f9396dc6b9e3542a6aa50a3;
2a375d2a9c41af31554bafb4a712097cc016d5227cb1f07652f0ef3483d5be30;
55cee457c73aa87258a04562c9d04cd3c865608d5dd64366d9cd9bc2fe2f5dd9;
a4cebac7bf4e5faa537a6013e9ae19c683d7cdad9dd318fdd968a966dd3a3010;
cb3039dad0ebd63e40fbcdbb8a2a1cdf9f442b2870383f5d469765387d0c8ec0;
d4cb58f6167b72764a216d0ce6281d2251f02a696060eb425c9782283422a828;
91d3a9c6de14197fe3be7c2b86b88b58b1f731d3e82bb0b7b11d5c75fbbed9a5;
b6ca1211159e9fd790790e49db5eb1b7a11c09f746d3135ae7a67ce8f518a403;
e18f051ac27ed29f792db49e4333adca9b1762d485a9214b5af12ffe858ca3fc;
381bcf2b7fefcdade08bb6a02dc32ea535dbef9cb9a43220649916db8bcc39d8;
502953496a40661bb6336a693371d3dd29ad96feb5e9f91a5b5ca0ad3ffbf29f;
52767ea5e20b8639433c087edf86ef91b0cb7fda46c71dcce625938a9f5d8a74;
4436c7024366356cd04724e1d6867786f2587a6f6295fc74b3af0c02a257adba;
4619cec6310e16d30e05204b35c084aabafabdd3d3f87661774fec253a103d11;
8eeab6635982618bebc137cf6c4795aa10010685d9c7bb6ce66932215195eed7;
92cd7309723461918b9cd2988a26cd2199749e82636dc6628a46878db7e12db3;
940a3ed18c4f171c9a6bccc0ab0ee8075aad6da8023e0b0e8883ca56bdddb4c7;
a348aabfd8aeec855933509c4c0b2aee78408ada89d8b51ce16b2247659b22f7;
ae35a7a1b084d09bb913b450944dc6f3205650298e58d19e3e2ee4db93a109ea;
b5ba8fbc4f5c9bbf01c9a0a533ecab0735bf8e5e63116fffc570392e6faa9d18;
b7666d4a0afe5f5b5de8faa541be31bbe34ea51c3b3a3fab77937f816ac6181e;
bbacf000880a46c7955a27f5dd960a6e253cd357f14f97f8472dd4fc3032f44d;
bda7ea39f9105c25250f14e9e1fa3de0f51b91b04349974c7cadbbbe1c06ce2f;
d2ccf6fa361ceaf8cebada53bb1f9458b016ad85b74a7dc1bf4ba18774d92645;
e7b59b841e127c6fe6e02dd98292bba49bd32350b57595e09a6adab8da78235b;
e810c74aefd63ce4ea674a1a961075a4d86a10b802d365b6b2b98a724d9b86db;
f467c72fa8adde6ddf27150122c117a17d1d664876c2f9d87e68e06257eb1904;
58b48fd39ef718e5bd501f57e83b537668b13176ca682aee36402d18bd0c0733;
59d880ae82ccc3c8207b745b1b3e55119a5b62af086a1639270b1ba5b7e1893a;
74d3093a51482a1eaa15e4fc8aa4b7d659d571db0570950272d7aa998aec6f49;
829b90bcf24fdf7f0298edec701c3c45b820f297dd012ac22e27e4bd295ee5f2;
9b6595980751537adf627e6107c08537de13e39752ed54c73e2b6af23e2a2769;
d711dc3c75a60ca0cd2556c267e3c33cee5d677edcfe70fb88b334f08f81ece9;
줄 3, 열 4
100%
Windows (CRLF)
UTF-8

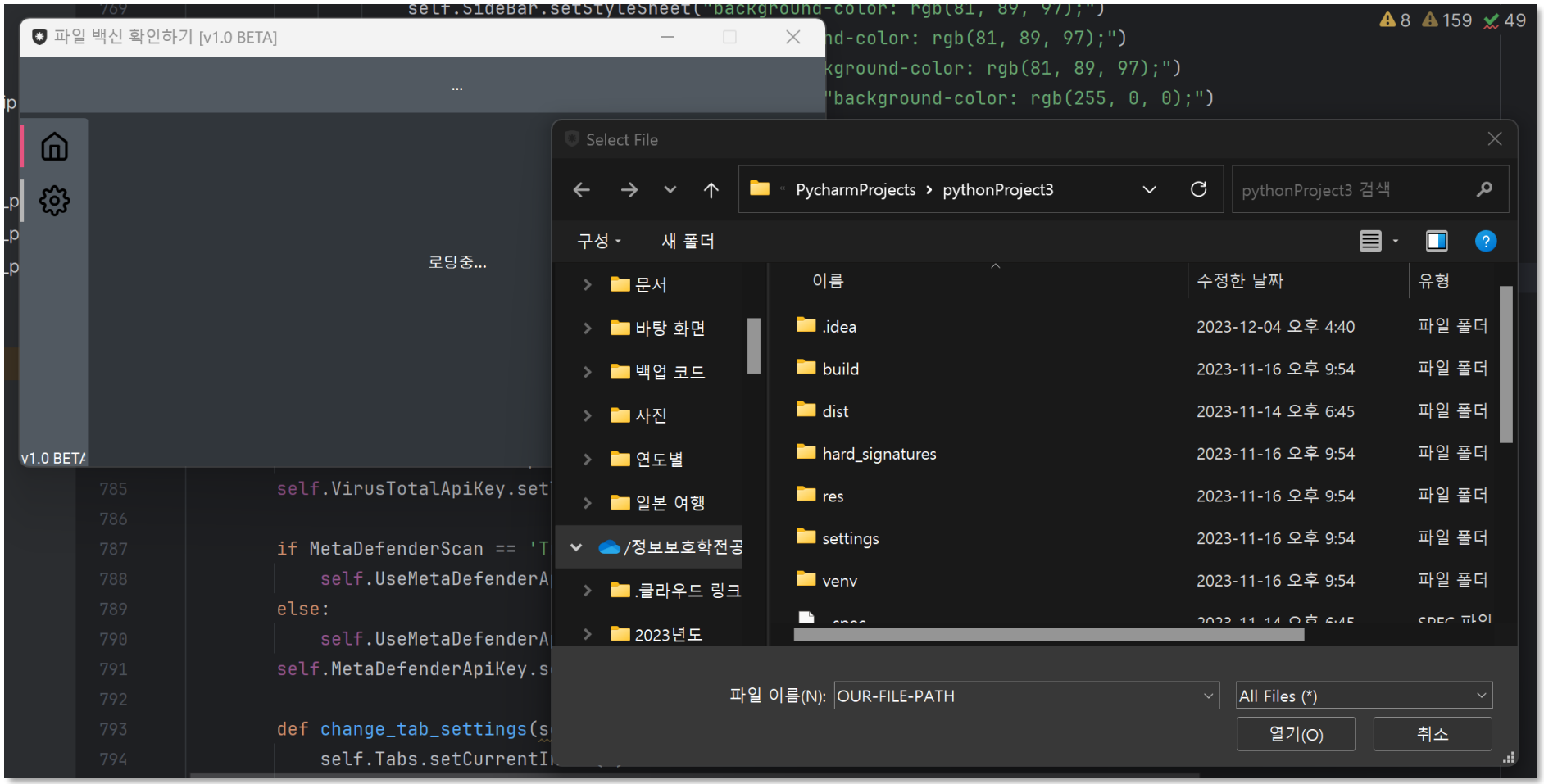
Anti-Virus Program 설명

실행 스크린샷_메인 및 설정 화면



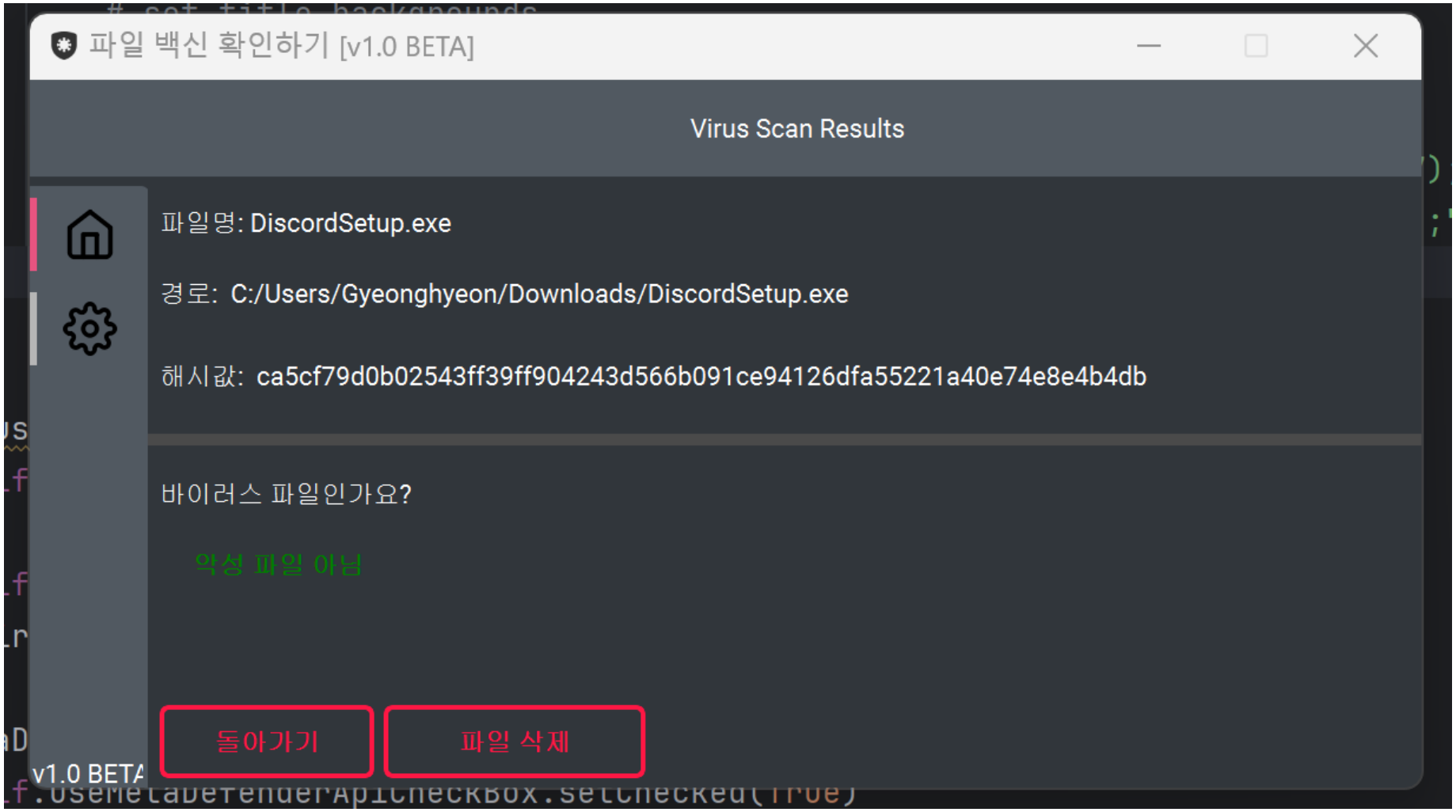
Anti-Virus Program 설명

실행 스크린샷_파일 설정 화면



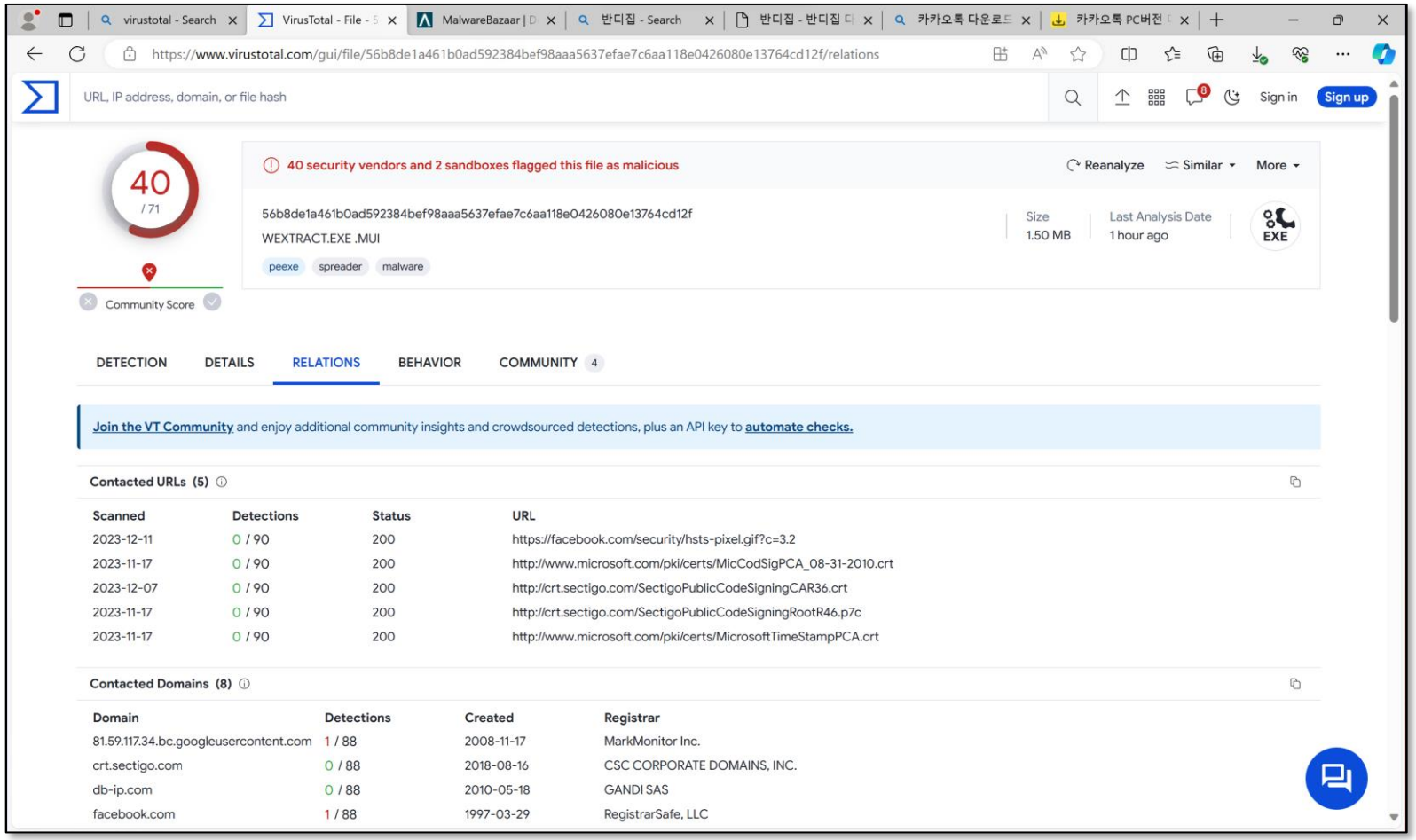
Anti-Virus Program 설명

실행 스크린샷_스캔 결과



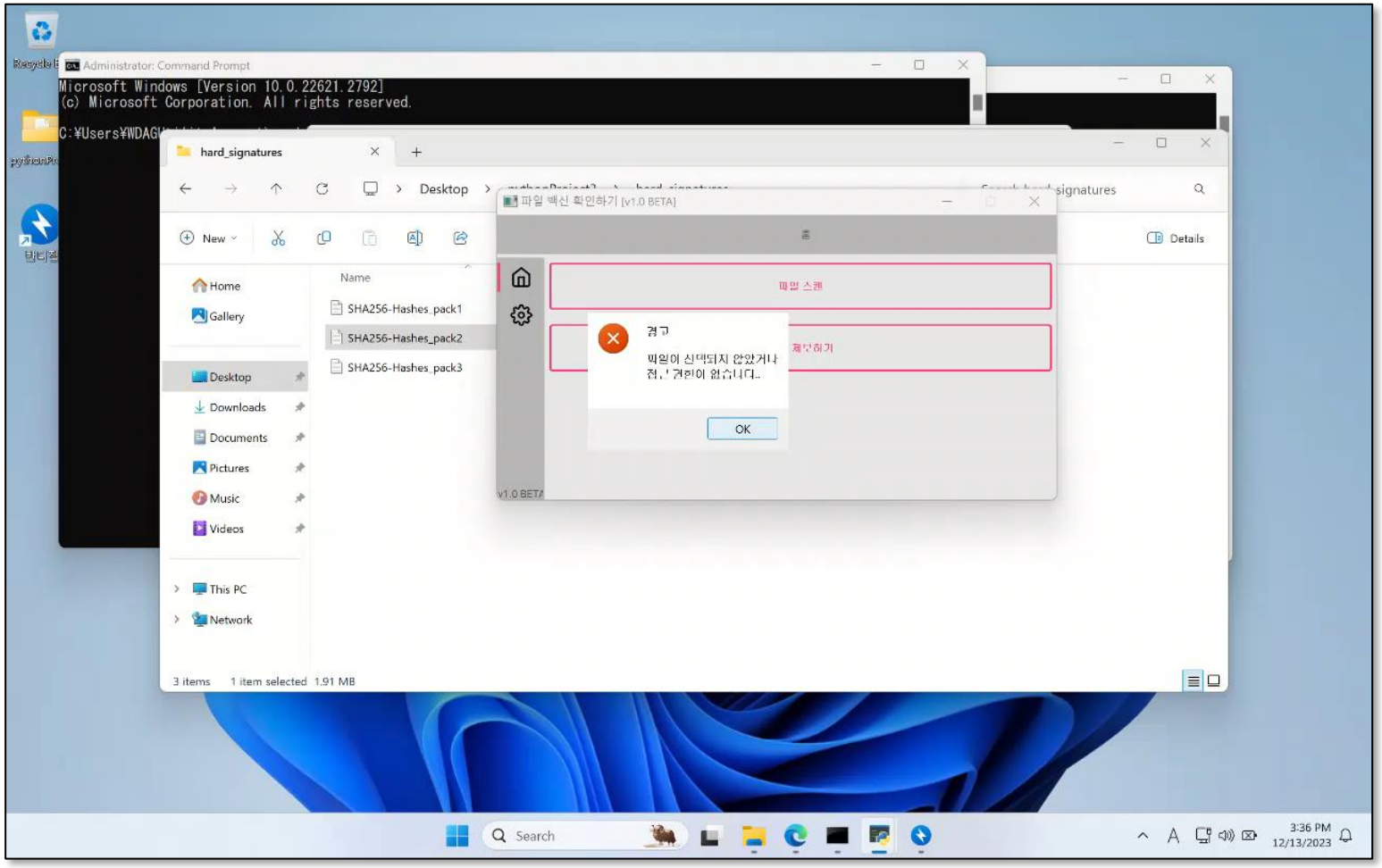
Anti-Virus Program 설명

실제 바이러스 기반 테스트



Anti-Virus Program 설명

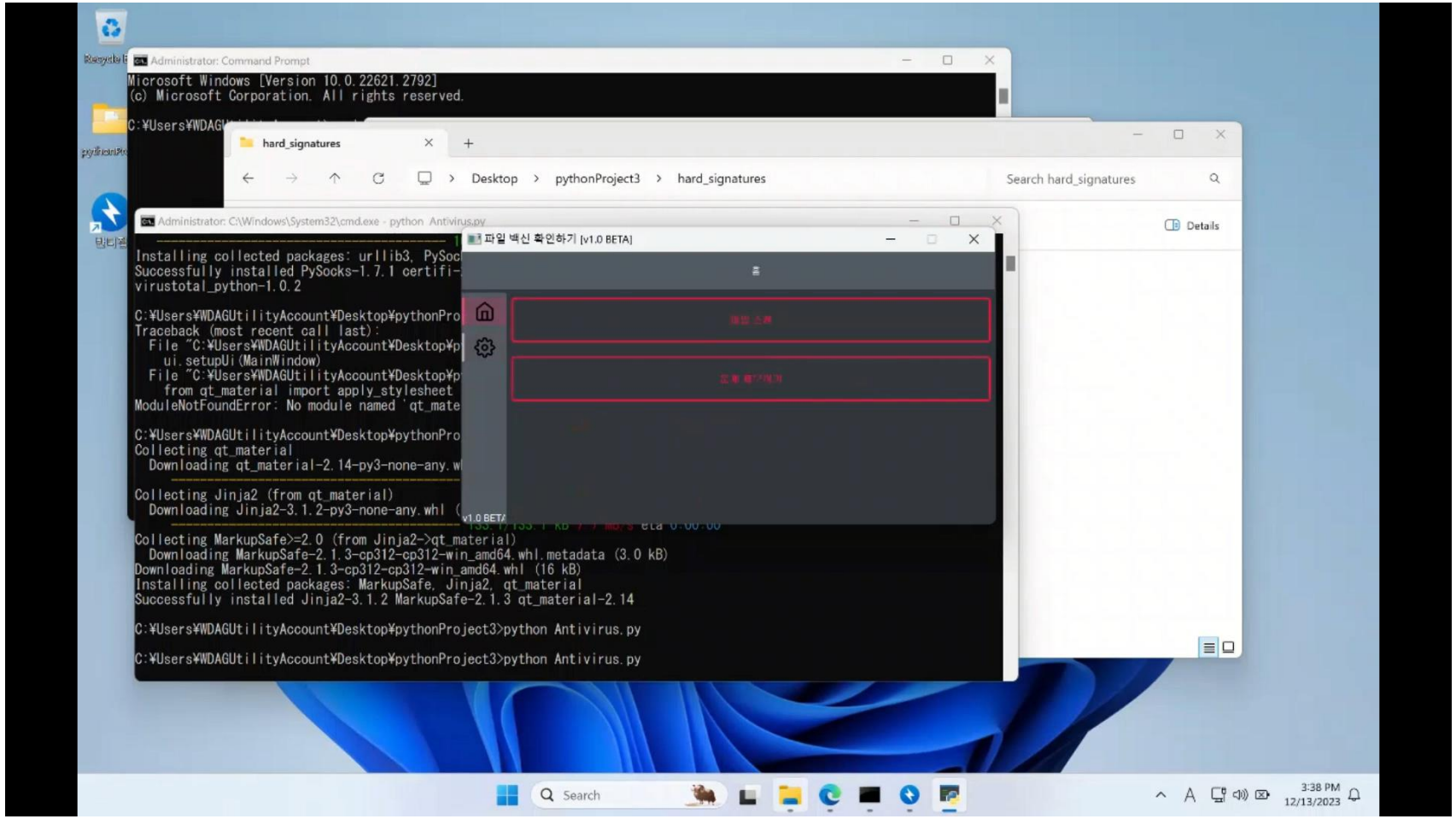
실제 바이러스 기반 테스트



정상 파일의 경우 바이러스 결과가 나타나지 않음

Anti-Virus Program 설명

실제 바이러스 기반 테스트



악성 파일의 경우 위와 같이 바이러스 감지 결과가 나옴

행위 기반 랜섬웨어 탐지_개념 설명

행위 기반 탐지란?

ㄴ 프로세스의 **행위를 분석**하여 랜섬웨어와 유사한 행위를 하는 **프로세스를 탐지**하는 방식

파일의 암호화

확장자나 배경화면 변경

랜섬웨어 유사한 행위를 하는 프로세스 탐지

행위를 미리 정의해야 하고, 행위를 미리 숨기거나 변형하는 경우 탐지하기 어려움

Anti-Virus Program 설명

행위 기반 랜섬웨어 탐지_개념 설명

Name	Date Modified	Size	Kind
demo	Today at 10:27 AM	30 KB	Folder
_rels	Today at 10:27 AM	588 bytes	Folder
.rels	Jan 1, 1980 at 12:00 AM	588 bytes	Unix executable
[Content_Types].xml	Jan 1, 1980 at 12:00 AM	1 KB	XML Document
docProps	Today at 10:27 AM	1 KB	Folder
app.xml	Jan 1, 1980 at 12:00 AM	795 bytes	XML Document
core.xml	Jan 1, 1980 at 12:00 AM	623 bytes	XML Document
xl	Today at 10:28 AM	20 KB	Folder
_rels	Today at 10:27 AM	698 bytes	Folder
workbook.xml.rels	Jan 1, 1980 at 12:00 AM	698 bytes	Document
sharedStrings.xml	Jan 1, 1980 at 12:00 AM	289 bytes	XML Document
styles.xml	Jan 1, 1980 at 12:00 AM	2 KB	XML Document
theme	Today at 10:27 AM	8 KB	Folder
theme1.xml	Jan 1, 1980 at 12:00 AM	8 KB	XML Document
workbook.xml	Jan 1, 1980 at 12:00 AM	2 KB	XML Document
worksheets	Today at 10:27 AM	1 KB	Folder
sheet1.xml	Jan 1, 1980 at 12:00 AM	1 KB	XML Document
demo.xlsx	Today at 10:26 AM	9 KB	Micros...k (.xlsx)

일부 차이는 있으나

공통된 구조를 가짐 !

행위 기반 랜섬웨어 탐지_코드 설명


```python
def generate_special_string(length):
    safe_special_characters = "!@#$%^&()"
    return ''.join(random.choice(safe_special_characters) for _ in range(length))


3 개의 사용 위치
def generate_decoy_file(folder, extension):
    excluded_characters = ["\"", "/", "\\", "|", "?", "*"]
    file_name = f"{generate_special_string(8)}"

    for char in excluded_characters:
        file_name = file_name.replace(char, __new: "")

    file_name = file_name + extension
    file_path = os.path.join(folder, file_name)

    with open(file_path, "w") as decoy_file:
        decoy_file.write("미끼 파일입니다. 수정하지 마세요 !")

    return file_path, file_name
```

```python
# pptx 생성
1개의 사용 위치
def generate_random_pptx(folder):
    prs = Presentation()
    title_slide_layout = prs.slide_layouts[0]
    slide = prs.slides.add_slide(title_slide_layout)
    title = slide.shapes.title
    subtitle = slide.placeholders[1]

    title.text = "Please do not remove this file"
    subtitle.text = "파일을 지우지 말아주세요."

    file_name = f"{generate_special_string(8)}.pptx"
    file_path = os.path.join(folder, file_name)

    prs.save(file_path)
    return file_path, file_name
```

Anti-Virus Program 설명

행위 기반 랜섬웨어 탐지_코드 설명


```python
class FileNameChangeHandler(FileSystemEventHandler):
    def __init__(self):
        self.log_file = "decoy_modified.txt"

    def on_moved(self, event):
        if event.is_directory:
            return

        file_path_before = event.src_path
        file_path_after = event.dest_path

        file_name_before = os.path.basename(file_path_before)
        file_name_after = os.path.basename(file_path_after)

        file_hash_before = get_file_hash(file_path_before) if os.path.exists(file_path_before) else None
        file_hash_after = get_file_hash(file_path_after) if os.path.exists(file_path_after) else None
```

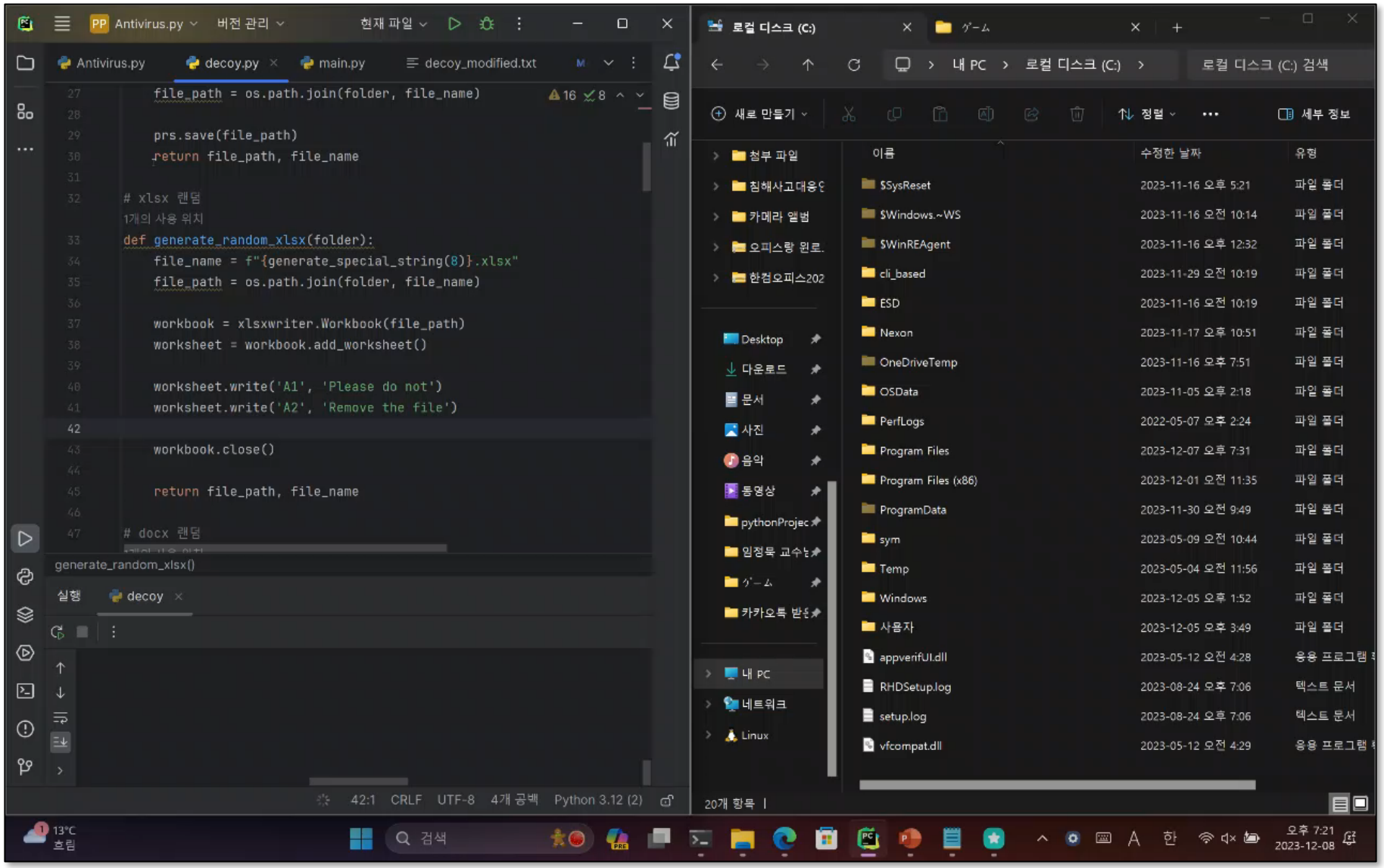
```python
if __name__ == "__main__":
    os.system("rmdir /s /q C:\\decoy")
    decoy_folder = "C:\\decoy"

    # Ensure the decoy folder exists
    if not os.path.exists(decoy_folder):
        os.makedirs(decoy_folder)

    decoy_files = [
        generate_random_docx(decoy_folder),
        generate_random_xlsx(decoy_folder),
        generate_random_pptx(decoy_folder)
    ]

    print(f"다음과 같은 디코이 파일을 생성되었습니다:")
    for file_path, _ in decoy_files:
        print(file_path)

    start_file_system_observer(decoy_folder)
```

행위 기반 랜섬웨어 탐지_코드 설명


```
if '_rels/.rels' not in file_list or 'docProps/core.xml' not in file_list:
    print(file_list)
    with open(self.log_file, "a") as log:
        log.write(f"[{time.strftime('%Y%m%d_%H:%M:%S')}]\n")
        log.write(f"변경 전 파일 이름: {file_name_before}\n")
        log.write(f"변경 되고 나서의 파일 이름: {file_name_after}\n")
        log.write(f"변경 전 경로: {file_path_before}\n")
        log.write(f"변경 후 경로: {file_path_after}\n")
        log.write(f"변경된 파일 해시값: {file_hash_after}\n\n")
        print(file_list)

    notification.notify(
        title="보안 경고",
        message="디코이 파일이 변경되었습니다 !!",
        timeout=10
    )

    print(f"Decoy File has been modified: {file_name_before} -> {file_name_after}")
    print(f"유형 : 프로그램 변경 예상")
    print(file_list)
else:
```

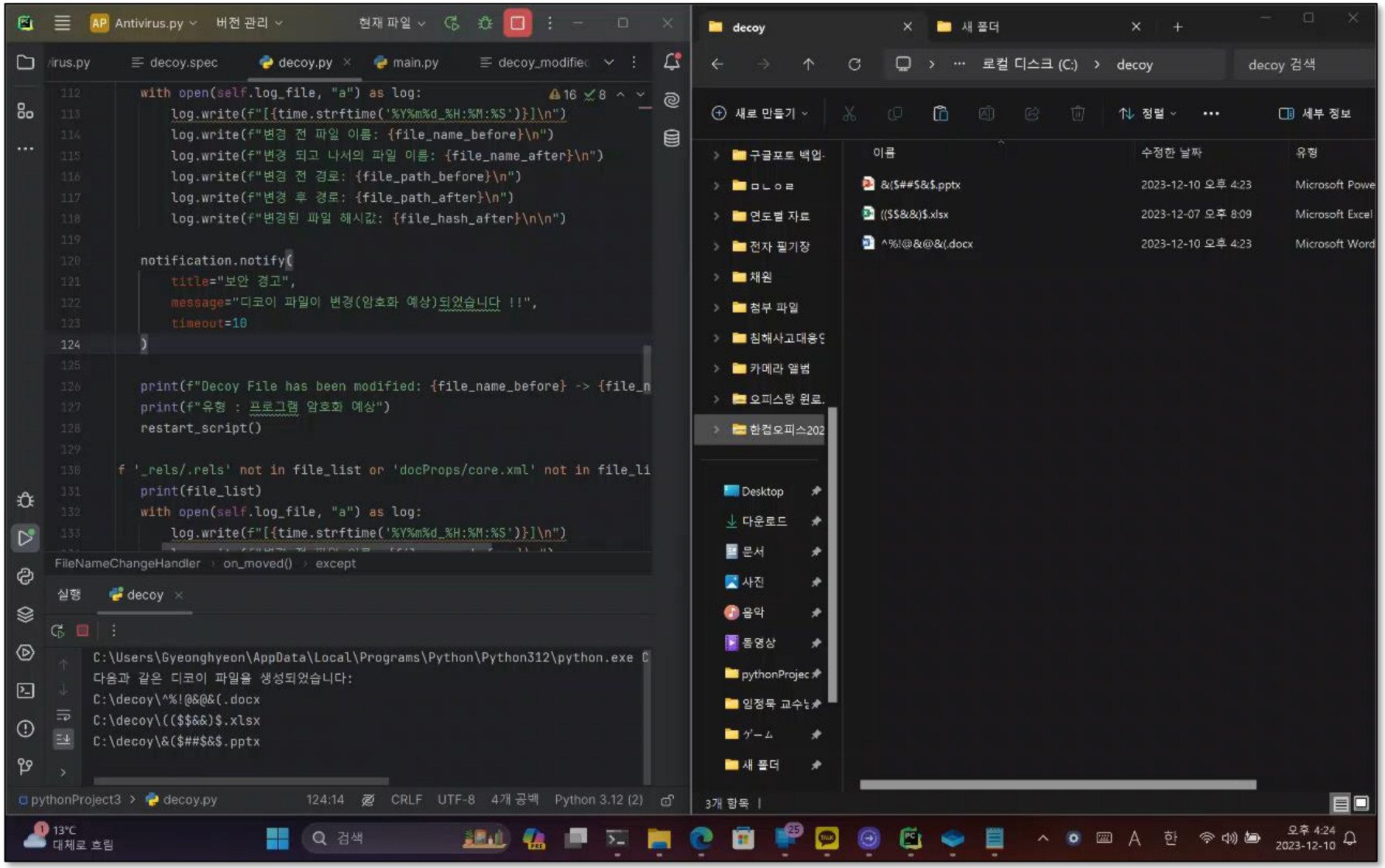
Anti-Virus Program 설명

행위 기반 랜섬웨어 탐지_작동 영상



Anti-Virus Program 설명

행위 기반 랜섬웨어 탐지_작동 영상



03 - 향후 계획 및 마무리

시도 및 진행중인 부분

실시간 검사 엔진 구축

- 시도 : 파이썬의 모듈 등만으로는 WinAPI 호출이 힘들었음.
  - 진행중인 부분 : 따라서 C++ 등의 언어로 개발 시도 예정

파일 검사 프로그램과 행위 기반 프로그램의 통합

- 시도 : 파일 검사 프로그램과 행위 기반 프로그램을 통합하였을 때에
  둘 중 하나가 랜덤으로 먹통되는 부분 존재
  - 진행중인 부분 : 백그라운드로 별도 실행 후 GUI 실행되게 조치 예정

행위 기반 프로그램의 예외 발생

- 시도 : 예상했던 것과는 다르게 타 프로그램에 의해 발생한 변경 또한 암호화로 인식함
  - 진행중인 부분 : WinAPI의 적극적인 활용으로 실행한 프로세스를 구분하는 엔진 개발

향후의 계획

Hash 외의 검증 방법 추가

➔ 현재는 3개의 방법 존재(hash, VirusTotal API, MetaDefender API)
➔ 추가할 수 있는 검증 방법 고안 必

엔진의 자동 업데이트 고안

➔ 현재는 수동으로 추가하거나, 사용자가 직접 Hash 파일을 수정하는 방식
➔ 서버에서 자동으로 이를 받아와서 각종 부분 업데이트 고안 必

Linux 작동 추가

➔ 현재는 Windows에서만 작동할 수 있는 코드 多
➔ 리눅스에서도 CLI 환경에서 작동할 수 있는 python 코드의 反映 필요


감 사 합 니 다