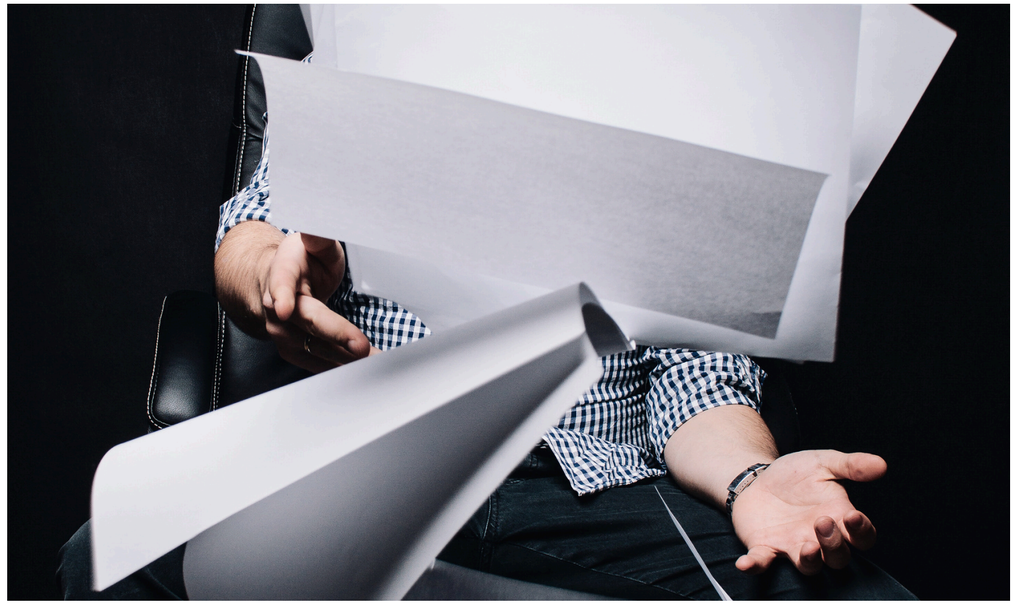
AI SECURITY

Ransomware Detection AI

정적 분석을 통한 랜섬웨어 탐지 및 복구 AI 개발



2024.10.25

발표자 : 김병민

PRESENTATION

네트워크 2팀

AGENDA

목차

01

프로젝트 주제 및 선정배경

프로젝트 주제

AI를 이용한 랜섬웨어 탐지 및 복구

주제 선정 이유

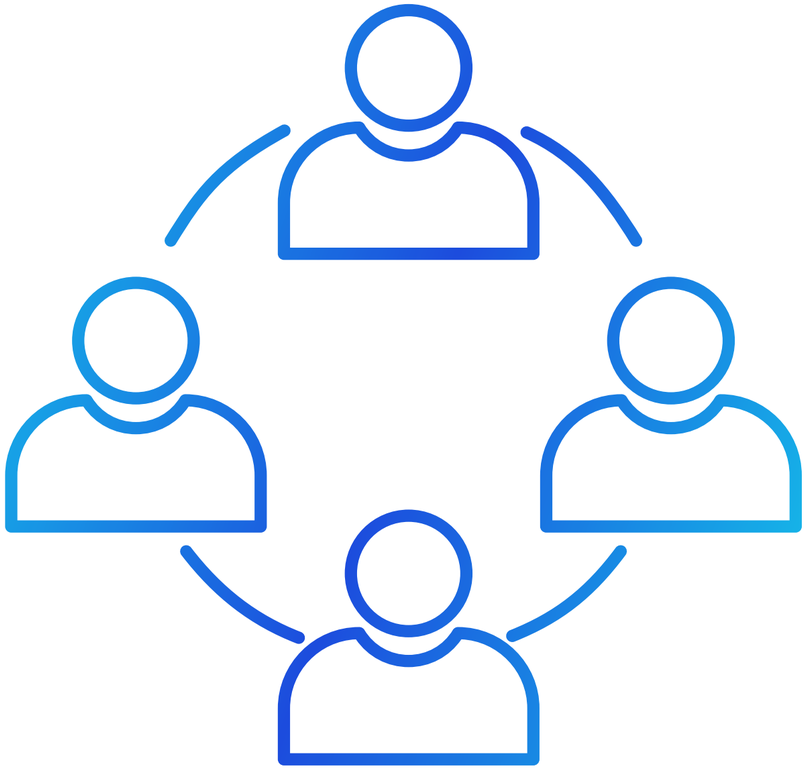
랜섬웨어에 의한 피해 증가를 AI 와 접목하여 해결하는 방안을 고안하게 됨

최종 목표

랜섬웨어 로그를 분석 후 LLM 모델을 활용하여 복호화 코드 작성

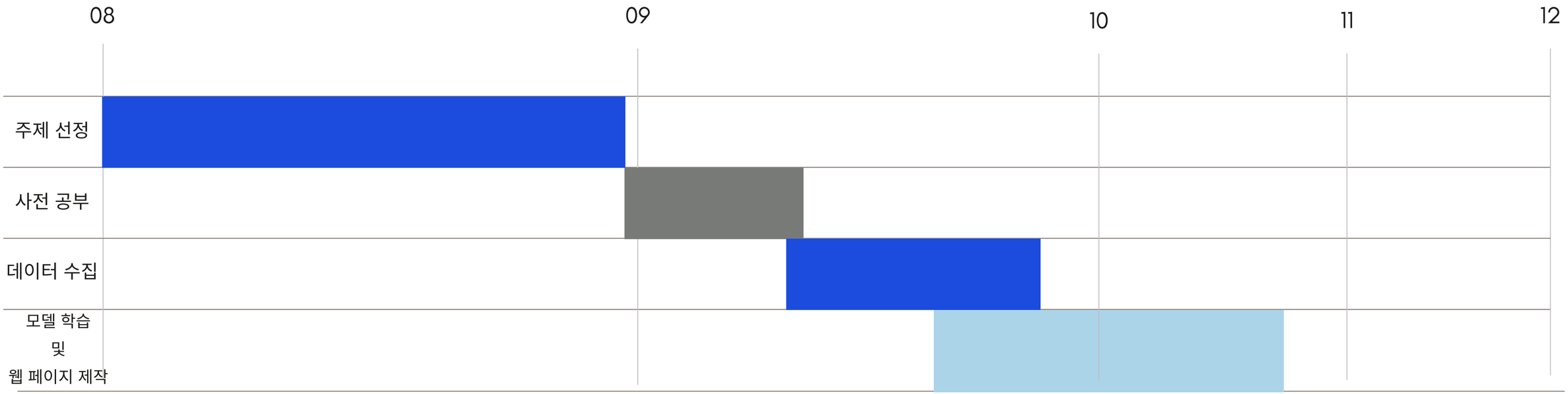
02

팀 소개



Mentor	누리랩 최원혁 대표님
서장석	정보보호학과 / 4학년
정채영	정보보호학과 / 4학년
이혜원	정보보호학과 / 1학년
이소정	스마트보안학부 / 3학년
김병민	컴퓨터정보공학과 / 2학년
이재호	스마트보안학부 / 2학년

03 프로젝트 추진 일정



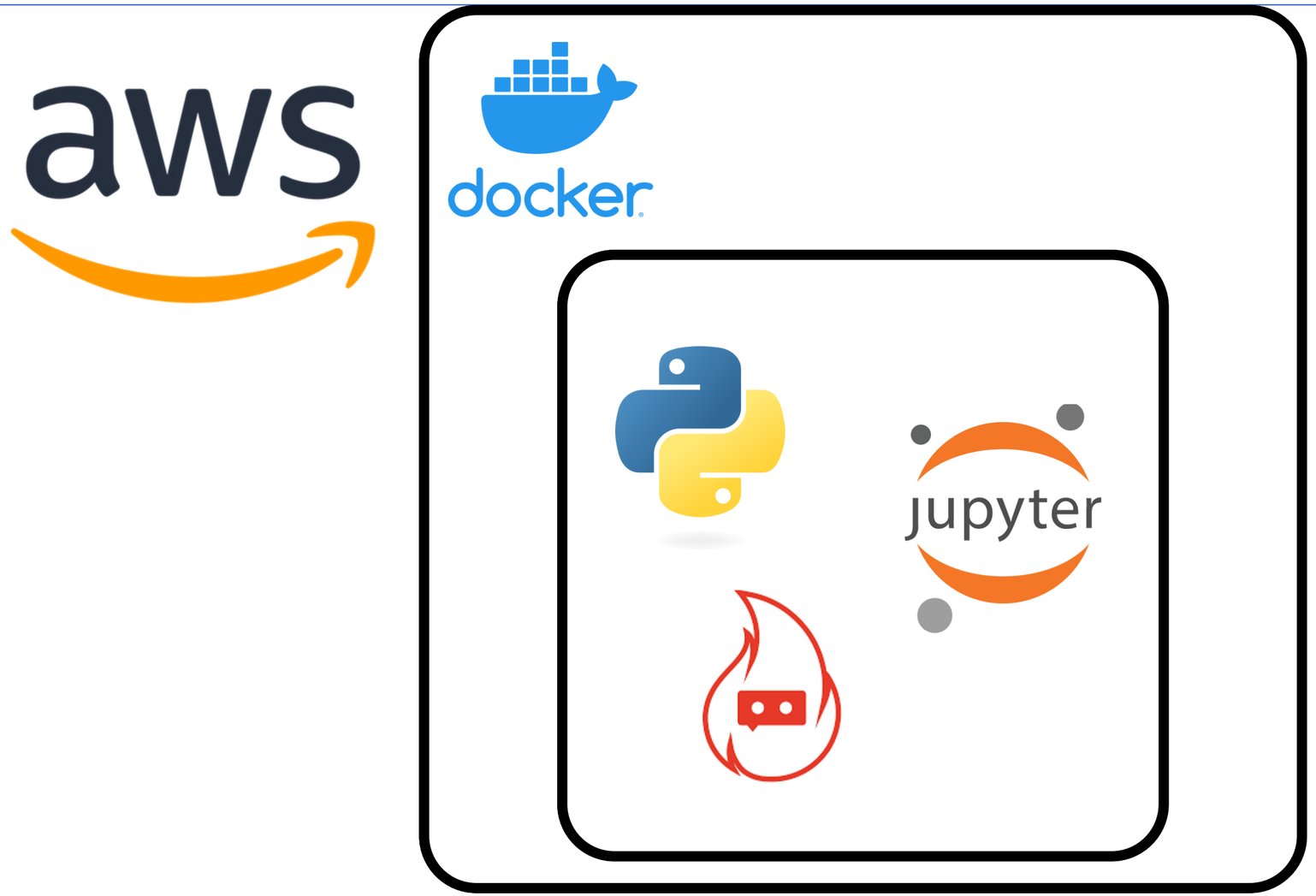
04 프로젝트 개발환경

Flask

web development,
one drop at a time

프로젝트 진행

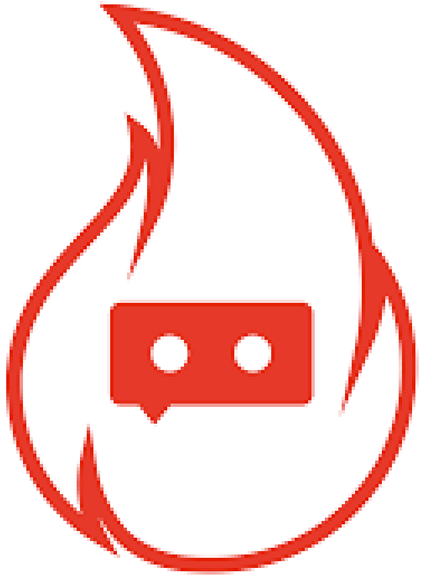
개발 환경 구축



05

프로젝트 진행

EMBER 프로젝트



Elastic Malware Benchmark for Empowering Researchers

악성코드 탐지를 위한 머신러닝 모델을 훈련시기기 위해 설계된 공개 데이터셋

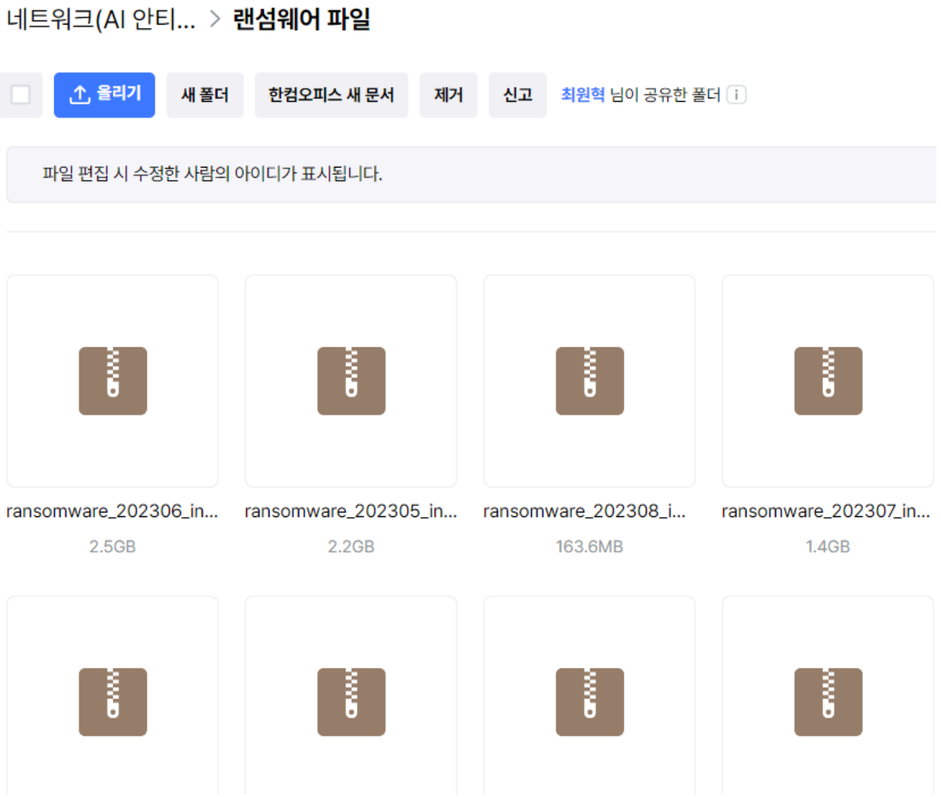
Windows의 실행 파일(PE 파일)의 정적 분석을 통한 악성코드 탐지 모델 개발에 사용

EMBER 프로젝트의 특징값 추출 방법과 기존 모델 파라미터를 이용 => 우리만의 새로운 모델 학습!

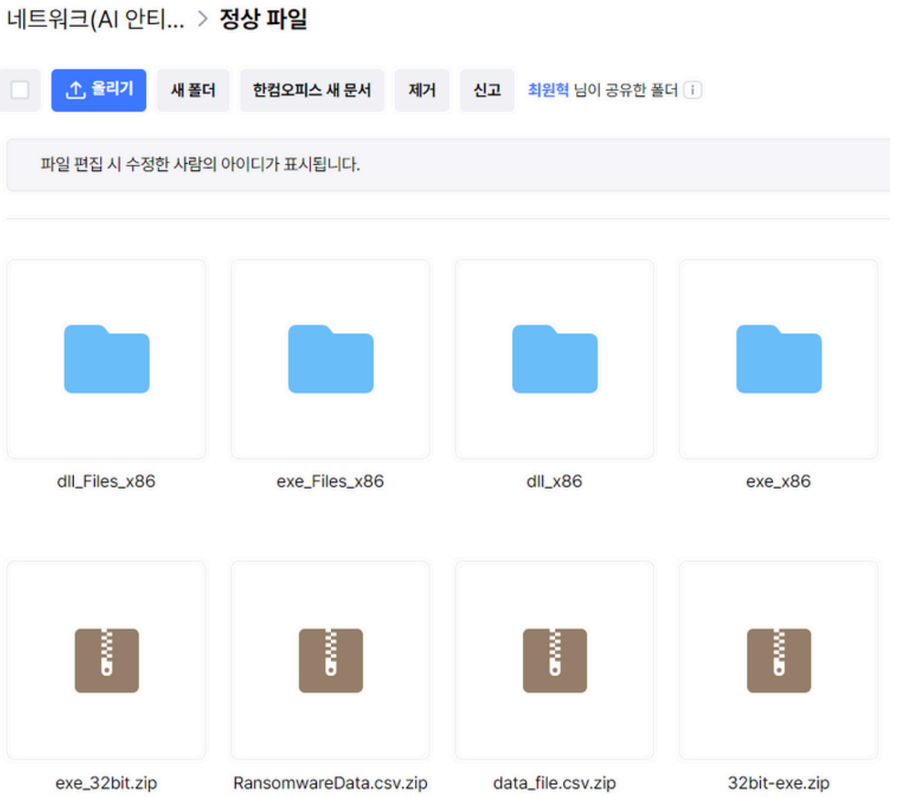
05

프로젝트 진행

32 bit 데이터 수집



랜섬웨어 파일



정상 파일

05

프로젝트 진행

모델 학습 과정

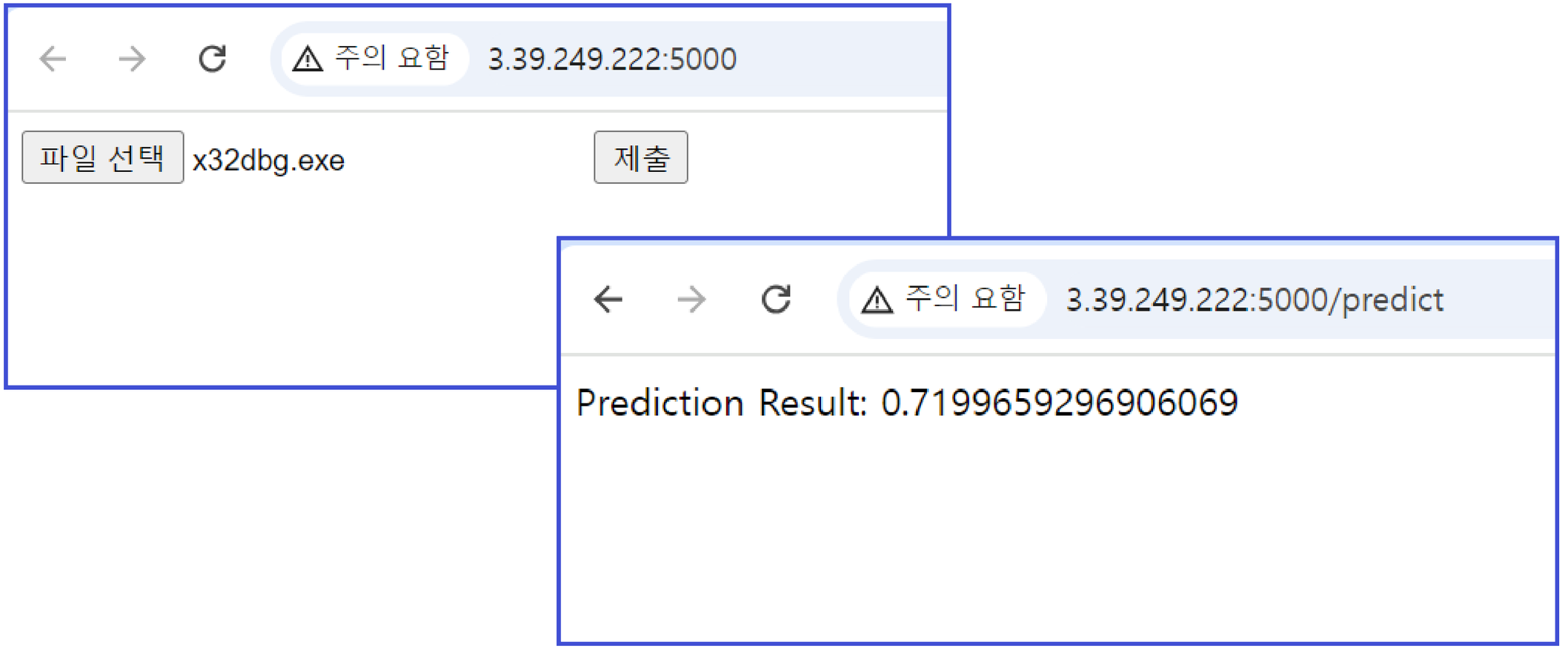
```
# 파일이 실제로 존재하는지 확인
if os.path.isfile(file_path):

    # 파일을 바이트 모드로 읽기
    with open(file_path, 'rb') as f:
        bytez = f.read()

    # 특징 추출
    feature_vector = extractor.feature_vector(bytez)
    features.append(feature_vector)

    # 파일명에 확장자가 있는지 확인
    _, file_extension = os.path.splitext(filename)
    if file_extension:
        # 확장자가 있으면 정상 파일 (0)
        labels.append(0)
        print(f"Processed normal file: {filename}")
    else:
        # 확장자가 없으면 랜섬웨어 (1)
        labels.append(1)
        print(f"Processed ransomware file: {filename}")
```

```
import lightgbm as lgb
from lightgbm import early_stopping
from sklearn.model_selection import train_test_split
import pandas as pd

# 데이터 로드
data = pd.read_csv('features.csv')

# 특징과 레이블 분리
X = data.drop('label', axis=1)
y = data['label']

# 학습 데이터와 검증 데이터 분리
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# LightGBM 데이터셋 생성
train_data = lgb.Dataset(X_train, label=y_train)
test_data = lgb.Dataset(X_test, label=y_test)

# 기존 EMBER 모델 로드
lgbm_model = lgb.Booster(model_file="ember_model_2018.txt")
params = lgbm_model.params
```

```
# early_stopping_round 양수 설정
params['early_stopping_round'] = 10

callbacks = [early_stopping(stopping_rounds=10)]

# 모델 학습
num_boost_round = 100
lgbm_model = lgb.train(
    params,
    train_data,
    num_boost_round=num_boost_round,
    valid_sets=[test_data],
    callbacks=callbacks
)
```

05 프로젝트 진행

모델 학습 결과 확인 및 평가



05

프로젝트 진행

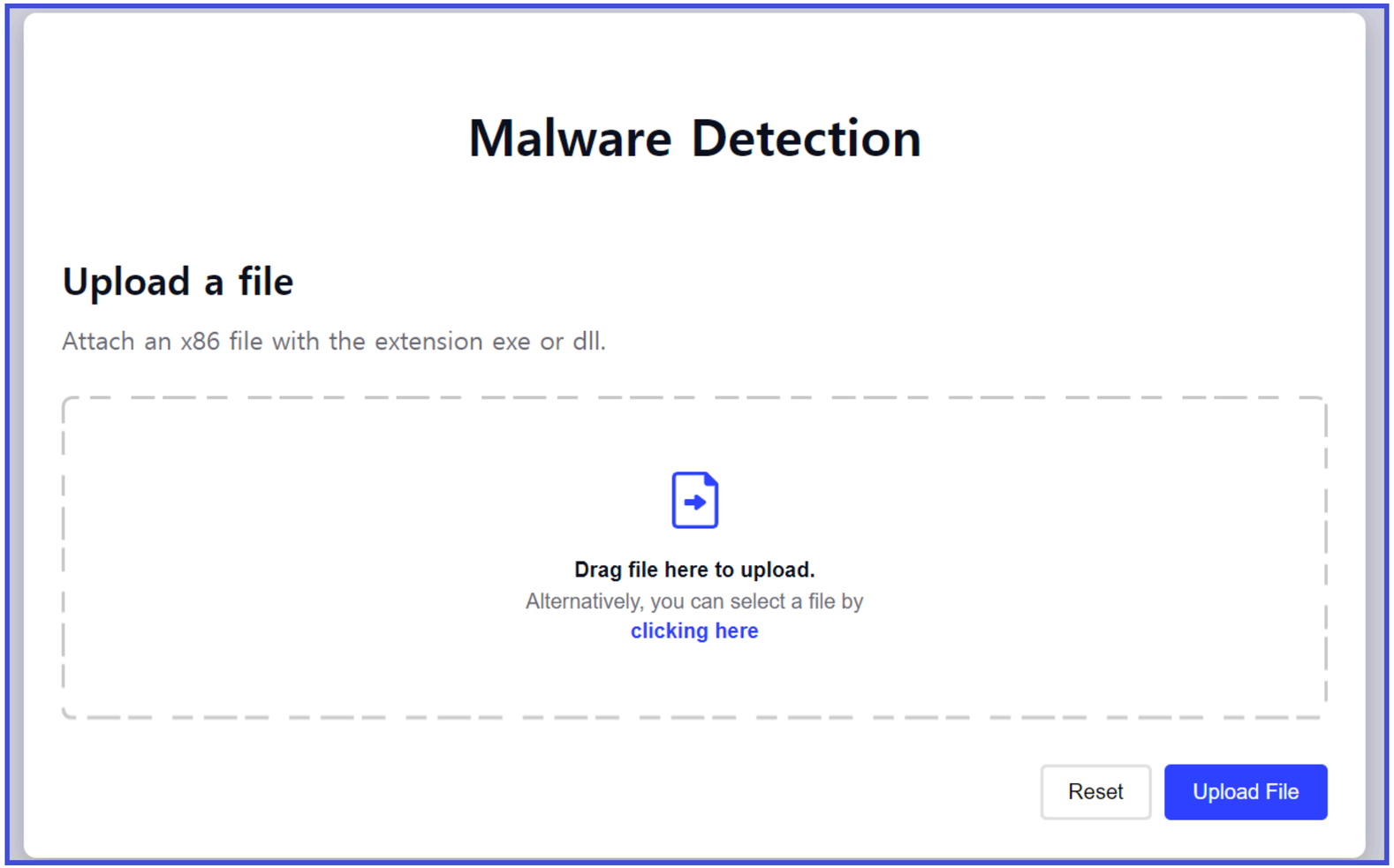
웹 서버 구현

File Edit View Language Python

```python
from flask import Flask, render_template, request, redirect, jsonify, session
import os
import tempfile
import ember
import lightgbm as lgb

app = Flask(__name__)

app.secret_key = 'QUIzcHhtZG5qemlxaGRrcw=='

# 파일 임시 저장 디렉토리 생성
UPLOAD_FOLDER = tempfile.gettempdir()
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# 모델 불러오기
lgbm_model = lgb.Booster(model_file="/home/ember_model_2024.txt")

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload', methods=['POST'])
def upload():

    if 'file' not in request.files:
        return redirect(request.url)

    file = request.files['file']

    if file.filename == '':
        return redirect(request.url)

    # 임시 파일 경로 생성
    temp_file_path = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)

    # 파일을 임시 디렉토리에 저장
    file.save(temp_file_path)

    # 세션에 파일 경로와 이름 저장
    session['temp_file_path'] = temp_file_path
    session['filename'] = file.filename

    if file:
```

06

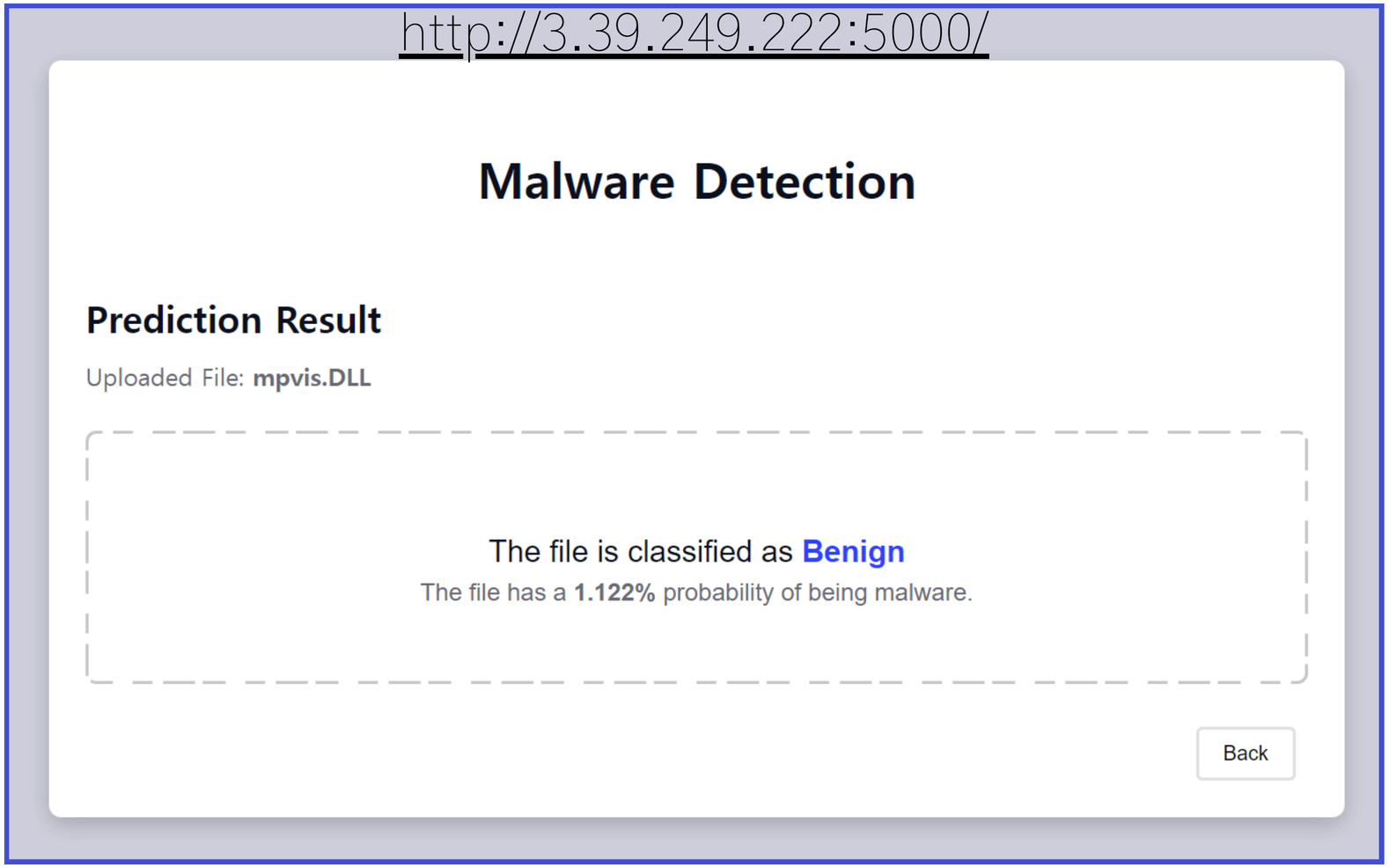
프로젝트 결과



1. 실행 파일을 드래그하여 업로드한다.

06 프로젝트 결과

http://3.39.249.222:5000/



Predict 결과가

80% 이상 – Malware

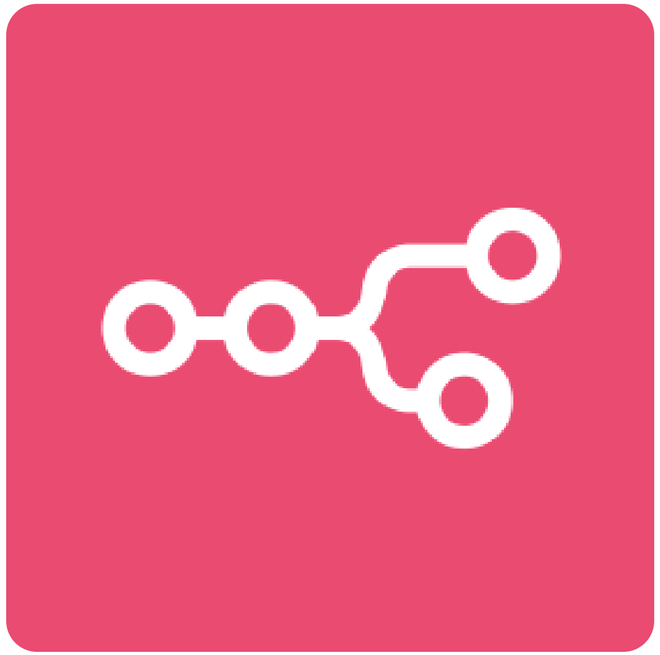
80% 미만 – Benign

2. 파일을 분석하여 랜섬웨어일 확률을 표시한다.

06

방향성 제시

워크플로우 자동화 툴을 통한 Open AI와의 연계



Open source Workflow Automation tool – n8n

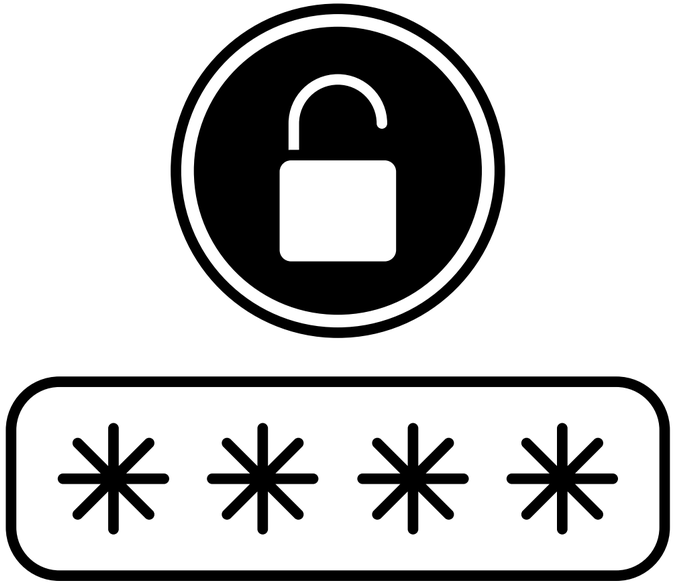
앞서 사용한 EMBER 모델을 다른 서비스와 연계하기 위해 사용

AI Agent들을 보다 적극적으로 이용하기에 용이한 툴

06

방향성 제시

복호화 코드 추천



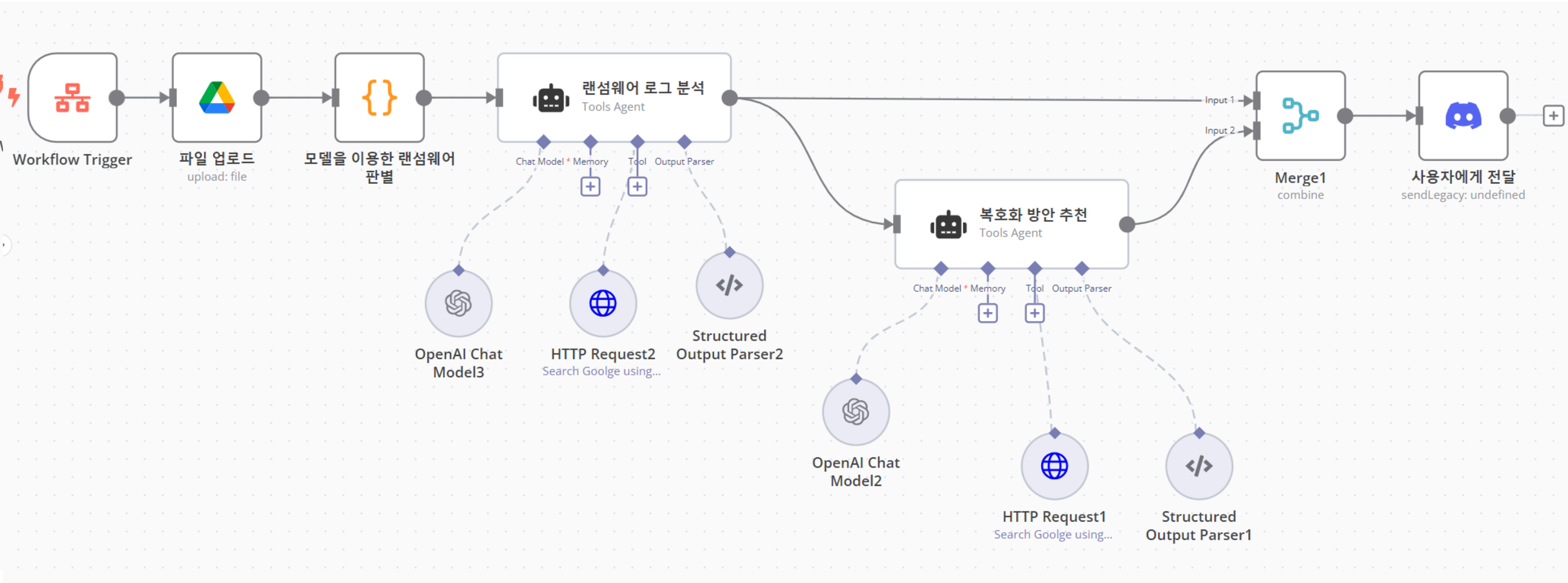
해당 랜섬웨어 로그 분석 후 복호화 코드 제시.

악성 파일의 종류를 분류하고 분류된 파일 특성에 맞게 준비된

복호화 코드를 제시함으로써 사용자가 랜섬웨어를 풀 수 있게 함.

분류된 로그 정보를 LLM 모델을 활용하여 복호화 코드 작성

06 방향성 제시


Workflow Trigger
파일 업로드
upload: file
모델을 이용한 랜섬웨어 판별
랜섬웨어 로그 분석
Tools Agent
Chat Model * Memory Tool Output Parser
OpenAI Chat Model3
HTTP Request2
Search Goolge using...
Structured Output Parser2
복호화 방안 추천
Tools Agent
Chat Model * Memory Tool Output Parser
OpenAI Chat Model2
HTTP Request1
Search Goolge using...
Structured Output Parser1
Input 1
Input 2
Merge1
combine
사용자에게 전달
sendLegacy: undefined

THANK YOU

감사합니다