

Anti_Pyrus 1st PROJECT

VACCINE PROGRAM with Python



CONTENTS

- 01 개요
- 02 활동내용
- 03 중간내용
- 04 결과
- 05 개선사항



프로젝트 개요

백신 개발을 선택한 이유와 목적을 설명합니다.

전체 흐름도

전체 흐름도를 통해 프로젝트의 전체 구성을 한눈에 보여줍니다.

01

OVERVIEW



프로젝트 개요

주제 선정 이유

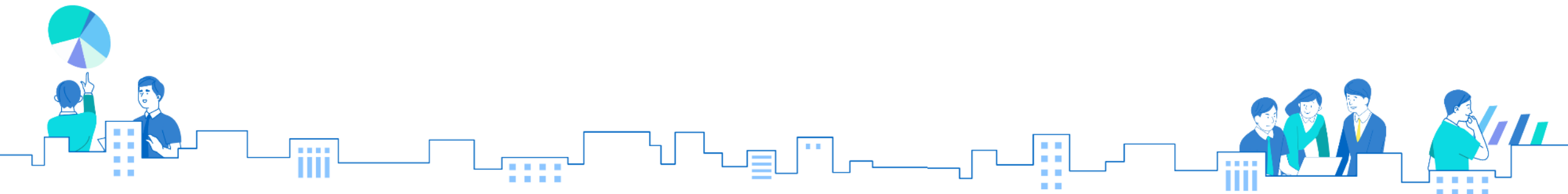
백신은 보안의 기본 형태이고

백신 개발은 이상징후 / 보안관제 / 이벤트 분석에 대해 포괄적으로 관련되어 있기에

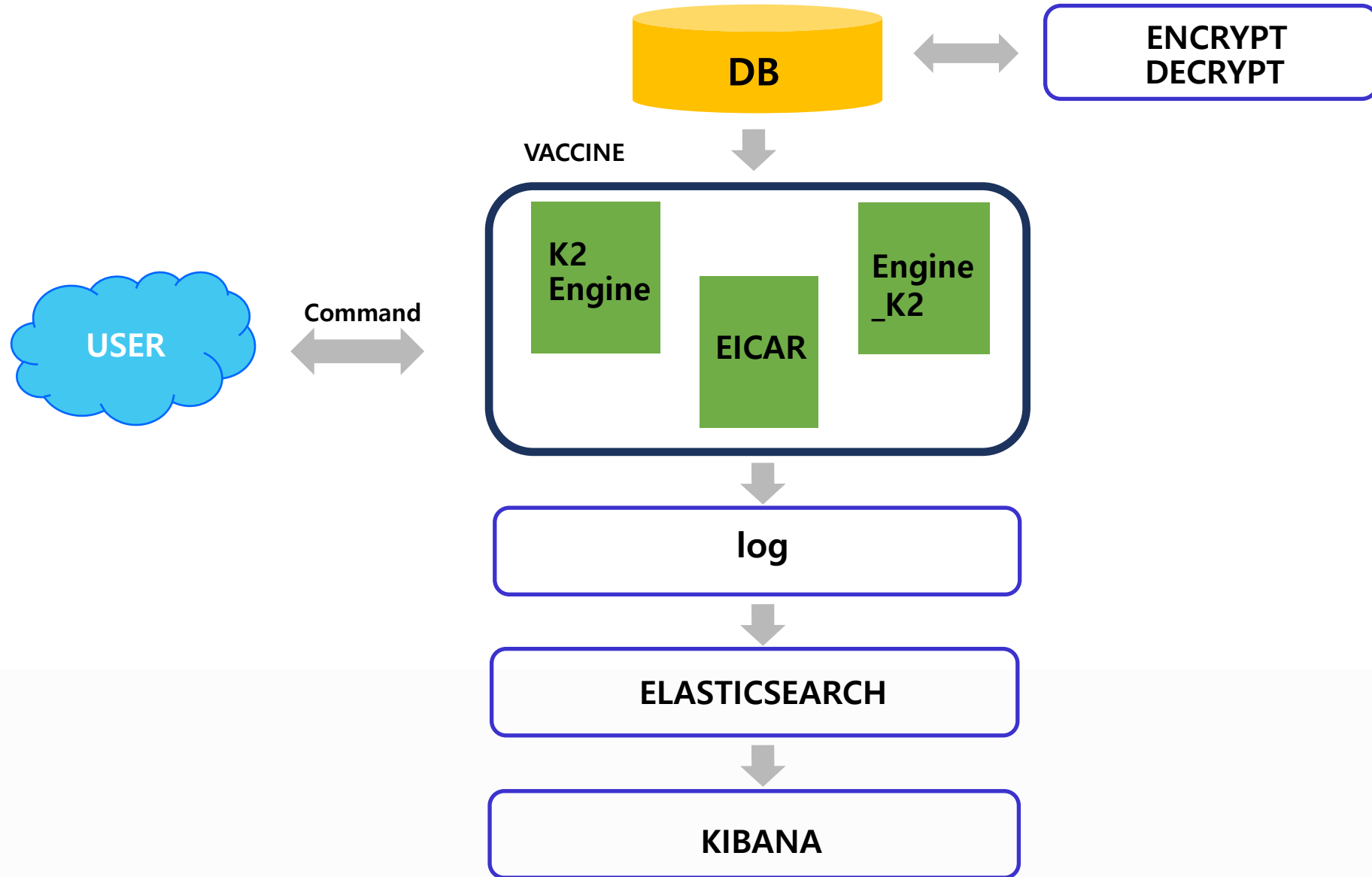
보안의 시작점이라고 생각하는 백신 개발을 선택

기대효과

1. 악성코드의 진단 원리를 이해할 수 있다.
2. 전체적인 백신의 구조와 동작 원리를 정확히 이해할 수 있다.
3. 로그 분석과 시각화에 대하여 학습할 수 있다.



전체 흐름도



팀원소개 및 담당업무

프로젝트 진행 인원을 소개합니다.

프로젝트 진행 기간

2022.11.04 ~ 2022.12.28

활동 내용

진행 중의 활동 내용과 더불어 프로젝트 기여도에 대한 설명입니다.

02

PROGRESS



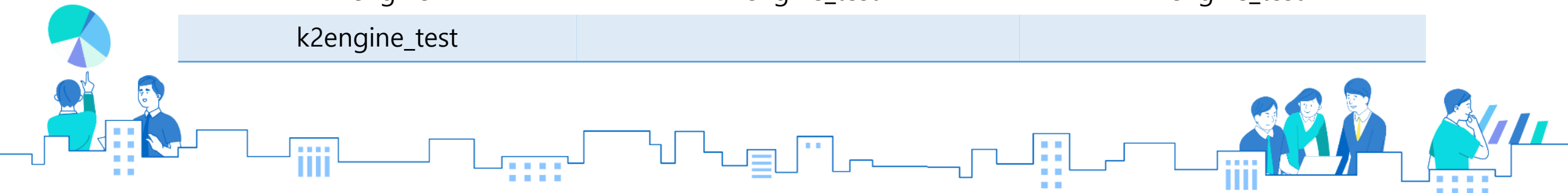
팀원소개 및 담당업무

조 한 비

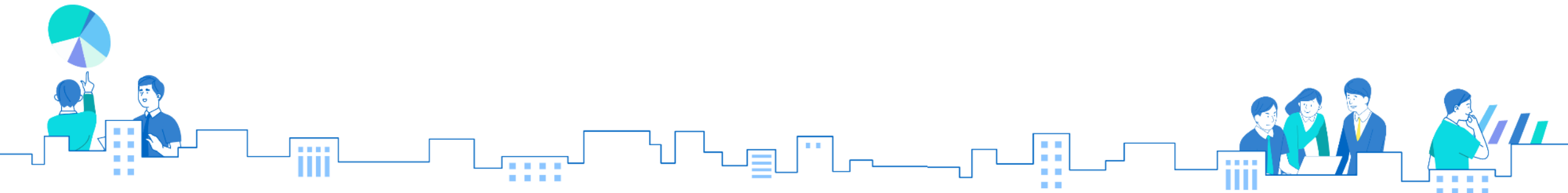
최 유 정

박 수 민

scanmod	antivirus	curemod
aes_decrypt	eicar	eicarmodule
aesencrypt	engine_k2	engine_k2
engine_k2	k2engine	k2engine
k2engine	k2engine_test	k2engine_test
k2engine_test		



프로젝트 기간



활동내용



백신 개발 파트분배

각자 파트 개발 후 github 로 형상관리
작업 후 발생하는 오류는 함께 해결
구동 테스트 진행



KIBANA 공동작업

포트연결(5601) 및 설치
데이터 쌓기
감염파일 개수 및 백신 엔진 가동시간 표시
Dashboard를 통한 그래프 시각화

ELASTICSEARCH

공동작업

포트연결(9200) 및 설치
Python Client 설치
외부접속이 가능하도록 설정변경
Mapping&Index 생성
색인화 과정

로그 생성

공동작업

로그 데이터 삽입
제이슨 타입 변환
지정값이 아닌 실제값으로 표시되도록 설정
로그를 보내기 위한 전송 모듈 생성



VACCINE

설계 구성도와 환경 스펙 및 소스코드의 중요한 부분을 보여줍니다.

ELASTICSEARCH

엘라스틱 서치의 전체 구성도와 설치 후의 과정을 상세적으로 보여줍니다.

KIBANA

백신 엔진 가동을 통해 나온 결과를 시각화를 위한 그래프화 하여 보여줍니다.

03

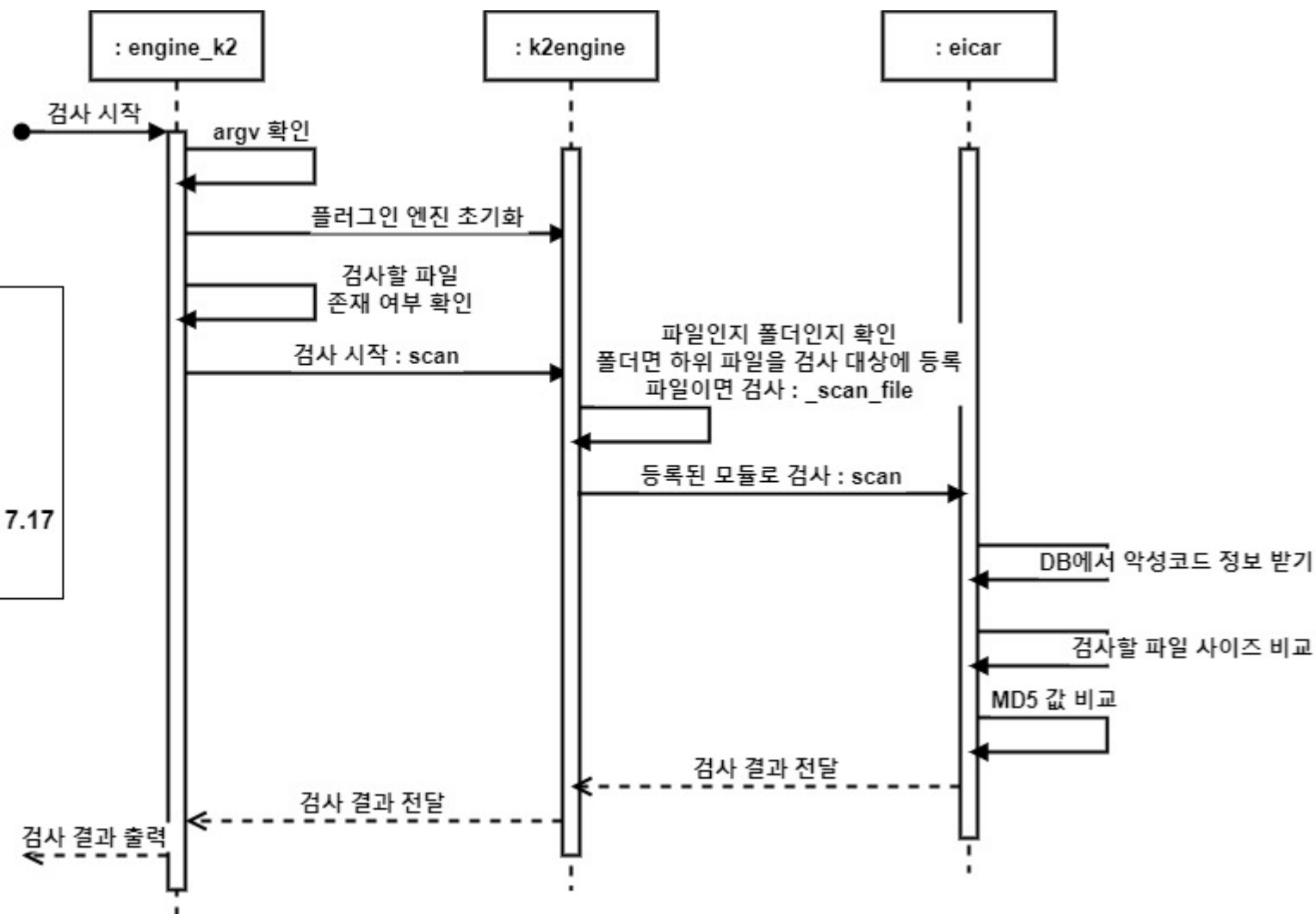
DEVELOPMENT



설계 구성도

개발 환경

- OS : windows 64bit
- CPU : Intel(R) core(TM) i7-10510U
- MEM: 32GB
- HDD : 690GB
- Develop Language : python 3.11
- 의존성 라이브러리 : elastic search client 7.17
- 형상관리 : Github



ENCRYPT



암호화된 파일을 로딩할 수 있도록 복호화

VACCINE 함수

함수 정리

init	엔진 초기화
uninit	엔진 종료
scan	엔진에 악성코드 검사 요청
disinfect	엔진에 악성코드 치료 요청
listvirus	엔진으로부터 진단 및 치료 가능한 악성코드 목록 획득
getinfo	엔진의 정보 획득



eicar / k2engine / engine_k2 소스코드 설계

VACCINE (EICAR+EICAR MODULE)

DISINFECT 함수

```
try:
    mm = filehandle
    size = os.path.getsize(filename)

    if vsize.count(size) :
        m = hashlib.md5()
        m.update(mm[:68])
        fmd5 = m.hexdigest()

        for t in vdb :
            if t[0] == fmd5 :
                return True, ' '+t[1], 0
```

Eicar.py의 scan함수 악성코드 검사

Size 확인한 후 같은 값이 있으면 md5 확인하여
악성코드 여부 판단

LOAD 함수

```
fp = open(mod_name + '.py', 'rb')
buf = fp.read()
fp.close()

module = importlib.import_module(mod_name)
exec(buf)
sys.modules[mod_name] = module
```

import module을 사용하여 Eicar를 module에 등록

VACCINE (K2ENGINE)

SCAN 함수

```
file_scan_list = [filename]

while len(file_scan_list) :
    try :
        real_name = file_scan_list.pop(0)

        if os.path.isdir(real_name) :
            if real_name[-1] == os.sep :
                real_name = real_name[:-1]
                ret_value['result'] = False
                ret_value['filename'] = real_name
                self.result['Folders'] += 1

            if self.options['opt_list'] :
                if isinstance(cb_fn, types.FunctionType) :
                    cb_fn(ret_value)

            flist = glob.glob(real_name + os.sep + '*')
            file_scan_list = flist + file_scan_list
```

파일인지 폴더인지 확인한 후

폴더라면, 폴더의 하위 파일을 검사대상 리스트에 등록

VACCINE (K2ENGINE)

SCAN 함수

```
elif os.path.isfile(real_name) :  
    ret, vname, mid, eid = self.__scan_file(real_name)  
  
    if ret :  
        self.result['Infected_files'] += 1  
        self.identified_virus.update([vname])  
  
    self.result['Files'] += 1  
  
    ret_value['result'] = ret  
    ret_value['engine_id'] = eid  
    ret_value['virus_name'] = vname  
    ret_value['virus_id'] = mid  
    ret_value['filename'] = real_name
```

CALLBACK 호출을 통한 파일의 악성코드 검사

RESULT 함수

```
def set_result(self) :  
    self.result['Folders'] = 0  
    self.result['Files'] = 0  
    self.result['Infected_files'] = 0  
    self.result['Identified_viruses'] = 0  
    self.result['IO_errors'] = 0  
  
def get_result(self) :  
    self.result['Identified_virus'] = len(self.identified_virus)  
    return self.result
```

set_result : 엔진의 악성코드 검사 결과 초기화

get_result : 엔진의 악성코드 검사 결과 출력

VACCINE (ENGINE_K2)

SCAN 함수

```
def scan_callback(ret_value) :  
    real_name = ret_value['filename']  
  
    disp_name = real_name  
    vname = ret_value['virus_name']  
  
    if ret_value['result'] :  
        state = 'infected'  
        message = state + vname  
  
    else :  
        message = 'ok'  
  
    elastic_log(real_name, vname)  
    display_line(disp_name, message)
```

악성코드가 발견되는 경우, 검사 대상 파일 이름 뒤에 infected 출력

```
for scan_path in args :  
    scan_path = os.path.abspath(scan_path)  
  
    if os.path.exists(scan_path) :  
        kav.scan(scan_path, scan_callback)  
    else :  
        print('Error : Invalid path: \'', scan_path, '\')
```

검사하려는 파일이 존재하는지 확인하고, scan 함수 호출

VACCINE (ENGINE_K2)

최종 악성코드 검사 결과

```
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\a.txt'
ok
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\b.txt'
ok
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\c.txt'
ok
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware1.txt'
infected EICAR-Test-File (not a virus)
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware2.txt'
infected EICAR-Test-File (not a virus)
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware_sample1.txt'
infected EICAR-Test-File (not a virus)
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware_sample2.txt'
infected EICAR-Test-File (not a virus)
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware_sample3.txt'
infected EICAR-Test-File (not a virus)
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware_sample4.txt'
infected EICAR-Test-File (not a virus)
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware_sample5.txt'
infected EICAR-Test-File (not a virus)
Results:
Folders      : 0
Files       : 10
Infected files : 7
Identified virus : 0
I/O errors  : 0
```



전체 폴더 개수

전체 파일 개수

악성코드 파일 개수

VACCINE

설계 구성도와 환경 스펙 및 소스코드의 중요한 부분을 보여줍니다.

ELASTICSEARCH

엘라스틱 서치의 전체 구성도와 설치 후의 과정을 상세적으로 보여줍니다.

KIBANA

백신 엔진 가동을 통해 나온 결과를 시각화를 위한 그래프화 하여 보여줍니다.

03

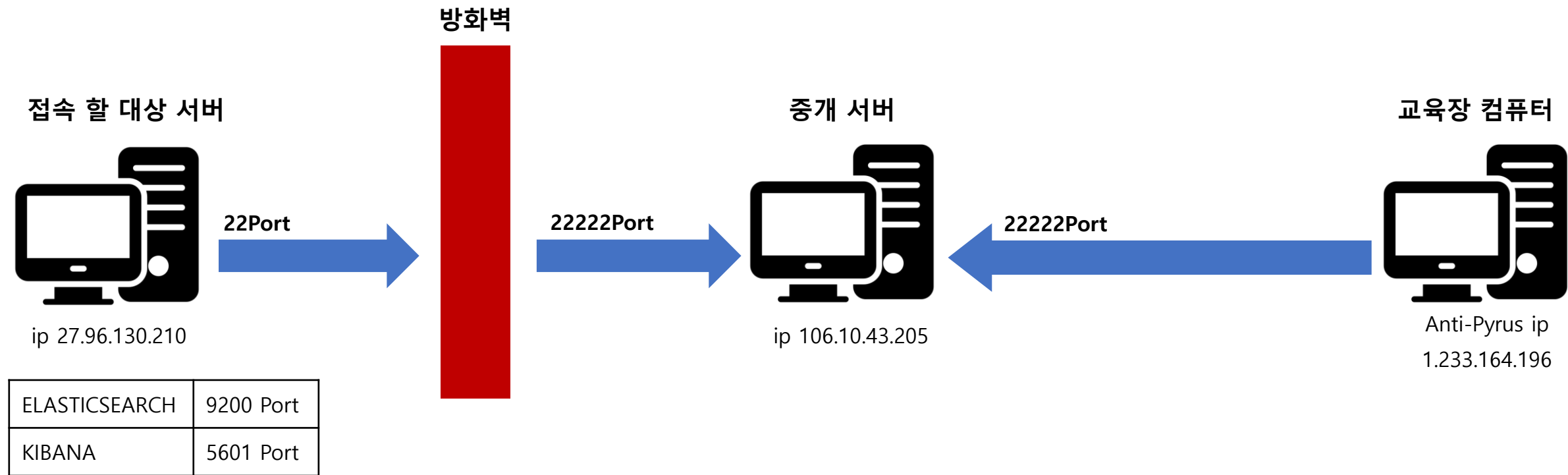
DEVELOPMENT



ELASTICSEARCH STRUCTURE



SSH PROTOCOL



서버 접속은 106.10.43.205를 통해 접근 가능하며, 외부 서비스는 공인 아이피를 통해 접근
Linux CentOS 7.8 64bit및 JAVA openjdk 1.8설치(CPU : 2Core / MEM : 8GB / SSD : 50GB)

ELASTICSEARCH INSTALL

Package	Arch	Version	Repository	Size
Installing:				
elasticsearch	x86_64	7.17.7-1	Elastic-7.x	300 M
Transaction Summary				
Install 1 Package				

Putty 이용해 접속 후 Elasticsearch install

```
C:\Users\Admin>pip install elasticsearch7
Collecting elasticsearch7
  Downloading elasticsearch7-7.17.7-py2.py3-none-any.whl (386 kB)
    386.2/386.2 kB 8.0 MB/s eta 0:00:00
Requirement already satisfied: urllib3<2,>=1.21.1 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from elasticsearch7) (1.26.13)
Requirement already satisfied: certifi in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from elasticsearch7) (2022.9.24)
Installing collected packages: elasticsearch7
Successfully installed elasticsearch7-7.17.7
```

Elasticsearch를 Python에서 사용할 수 있도록
Python Elasticsearch Client 7.17.7ver installation

ELASTICSEARCH PORT

외부에서 elasticsearch에 접근할 수 있도록 설정 변경

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
PID/Program name					
tcp	0	0	127.0.0.1:9300	0.0.0.0:*	LISTEN
7694/java					
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
847/sshd					
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN
521/rpcbind					
tcp	0	0	0 127.0.0.1:9200	0.0.0.0:*	LISTEN
7694/java					
tcp6	0	0	:::22	:::*	LISTEN
847/sshd					
tcp6	0	0	:::111	:::*	LISTEN
521/rpcbind					



Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
PID/Program name					
tcp	0	0	0.0.0.0:9300	0.0.0.0:*	LISTEN
12526/java					
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
847/sshd					
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN
521/rpcbind					
tcp	0	0	0 0.0.0.0:9200	0.0.0.0:*	LISTEN
12526/java					
tcp6	0	0	:::22	:::*	LISTEN
847/sshd					
tcp6	0	0	:::111	:::*	LISTEN
521/rpcbind					

로그 생성 후 JSON 타입 변환

시간 지정 -> realtime

file, file path -> 실제이름, 실제 경로

pc 및 ip 지정 값 -> 실제 pc 및 ip 값

username 값 출력

virus detection -> 실제 바이러스 탐지 시간

vname -> 실제 바이러스명

```
def elastic_log(real_name, vname) :  
    current_time = datetime.now()  
    current_time = current_time.strftime('%Y-%m-%d %H:%M:%S')  
    now_file = os.path.split(real_name)[-1]  
    ex_ip = socket.gethostbyname(socket.getfqdn())  
    pcname = socket.gethostname()  
    usr_list = {'DESKTOP-EF2BM5I' : 'user1', 'DESKTOP-3P32NCE' : 'user2', }  
    username = 'user4'  
    if pcname in usr_list :  
        username = usr_list[pcname]  
  
    log = {  
        "detection time" : current_time,  
        "field" : "virus",  
        "file" : now_file,  
        "file path" : real_name,  
        "hostname" : "antipyrus",  
        "ip" : ex_ip,  
        "pc" : pcname,  
        "username" : username,  
        "virus detection" : vname}  
  
    two.vaccine_anti(log)
```

elastic_log를 json타입으로 생성한 후 vaccine_anti 호출

ELASTICSEARCH MAPPING & INDEX

매핑 정보 조회

```
import json
from elasticsearch7 import Elasticsearch, helpers

_ES_URL = "27.96.130.210:9200"
_ES_INDEX = "antipy_log"
_DOC_TYPE = _ES_INDEX
es_client = Elasticsearch(_ES_URL, timeout=60*1)

with open("mapping.json", "r") as f:
    mapping = json.load(f)

es_client.indices.create(index=_ES_INDEX, body=mapping)
```

mapping 후 index 생성 확인



```
[root@dev-test1 ~]# curl -XGET 'http://localhost:9200/antipy_log/_mapping?pretty'
{
  "antipy_log" : {
    "mappings" : {
      "properties" : {
        "detection time" : {
          "type" : "date",
          "format" : "yyyy-MM-dd HH:mm:ss||yyyy-MM-dd||epoch_millis"
        },
        "field" : {
          "type" : "keyword"
        },
        "file" : {
          "type" : "text"
        },
        "file path" : {
          "type" : "text"
        },
        "hostname" : {
          "type" : "text"
        },
        "ip" : {
          "type" : "ip"
        },
        "pc" : {
          "type" : "text"
        },
        "username" : {
          "type" : "text"
        },
        "virus detection" : {
          "type" : "keyword"
        }
      }
    }
  }
}
```

전송모듈 생성

```
from elasticsearch7 import Elasticsearch, helpers

def vaccine_anti(log) :
    _ES_URL = "27.96.130.210:9200"
    _ES_INDEX = "antipy_log"
    es_client = Elasticsearch(_ES_URL, timeout=60*1)

    es_client.index(index=_ES_INDEX, doc_type=__doc__, body=log)
```

Elasticsearch로 로그를 보내기 위한 전송모듈

```
{
  "detection time" : "2022-12-07",
  "field" : "virus",
  "file" : "malware1",
  "file path" : "C:\\\\users\\\\Admin\\\\Desktop\\\\virus_project\\\\test",
  "hostname" : "antipyus",
  "ip" : "127.0.0.1",
  "pc" : "desktop",
  "username" : "admin",
  "virus detection" : "malware"
}
```

전송 확인을 위한 테스트 로그

VACCINE

설계 구성도와 환경 스펙 및 소스코드의 중요한 부분을 보여줍니다.

ELASTICSEARCH

엘라스틱 서치의 전체 구성도와 설치 후의 과정을 상세적으로 보여줍니다.

KIBANA

백신 엔진 가동을 통해 나온 결과를 시각화를 위한 그래프화 하여 보여줍니다.

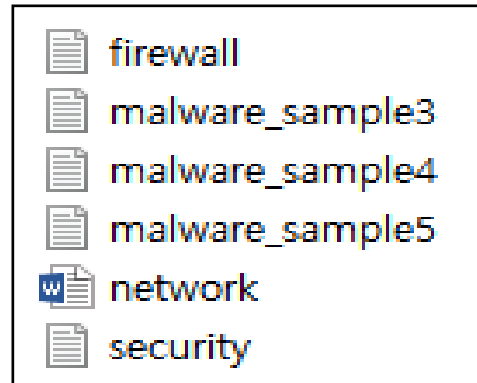
03

DEVELOPMENT



KIBANA TEST

엔진 가동 결과



바이러스 파일 : 3개
정상 파일 : 3개



```
b'C:\\Users\\Admin\\Desktop\\virus_project\\virustest'  
ok  
b'C:\\Users\\Admin\\Desktop\\virus_project\\virustest\\firewall.txt'  
ok  
b'C:\\Users\\Admin\\Desktop\\virus_project\\virustest\\malware_sample3.txt'  
infected malware_sample3  
b'C:\\Users\\Admin\\Desktop\\virus_project\\virustest\\malware_sample4.txt'  
infected malware_sample4  
b'C:\\Users\\Admin\\Desktop\\virus_project\\virustest\\malware_sample5.txt'  
infected malware_sample5  
b'C:\\Users\\Admin\\Desktop\\virus_project\\virustest\\network.docx'  
ok  
b'C:\\Users\\Admin\\Desktop\\virus_project\\virustest\\security.txt'  
ok  
Results:  
  
Folders      : 1  
Files        : 6  
Infected files : 3  
Identified virus : 0  
I/O errors   : 0  
6.09076 sec
```

바이러스 3개 검출

엔진 가동 시간 표시

KIBANA TEST 결과

KIBANA JSON 로그

```
> 2022-12-23 @ 11:13:25.000 detection time: 2022-12-23 @ 11:13:25.000 field: non-virus file: network.docx file path: C:\Users\Admin\Desktop\virus_project\virustest\network.docx hostname: antipyrus
ip: 192.168.56.1 pc: DESKTOP-EF2BM5I username: user1 virus detection: (empty) _id: yjvBPIUBrIUju-iNt4WU _index: antipy_log _score: - _type: _doc

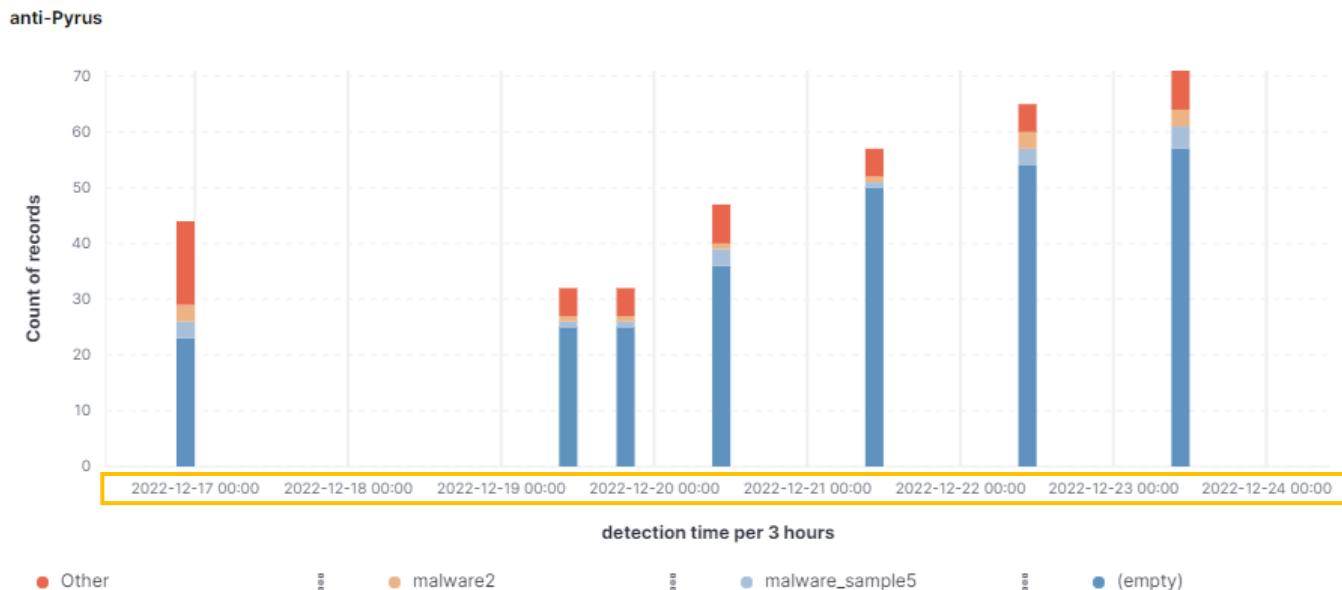
> 2022-12-23 @ 11:13:24.000 detection time: 2022-12-23 @ 11:13:24.000 field: non-virus file: mongshell.hwp file path: C:\Users\Admin\Desktop\virus_project\virustest\mongshell.hwp hostname: antipyrus
ip: 192.168.56.1 pc: DESKTOP-EF2BM5I username: user1 virus detection: (empty) _id: yTvBPIUBrIUju-iNtIUk _index: antipy_log _score: - _type: _doc

> 2022-12-23 @ 11:13:23.000 detection time: 2022-12-23 @ 11:13:23.000 field: virus file: malware_sample4.txt file path: C:\Users\Admin\Desktop\virus_project\virustest\malware_sample4.txt hostname: antipyrus
ip: 192.168.56.1 pc: DESKTOP-EF2BM5I username: user1 virus detection: malware_sample4 _id: xzvBPIUBrIUju-iNrYVM _index: antipy_log _score: - _type: _doc

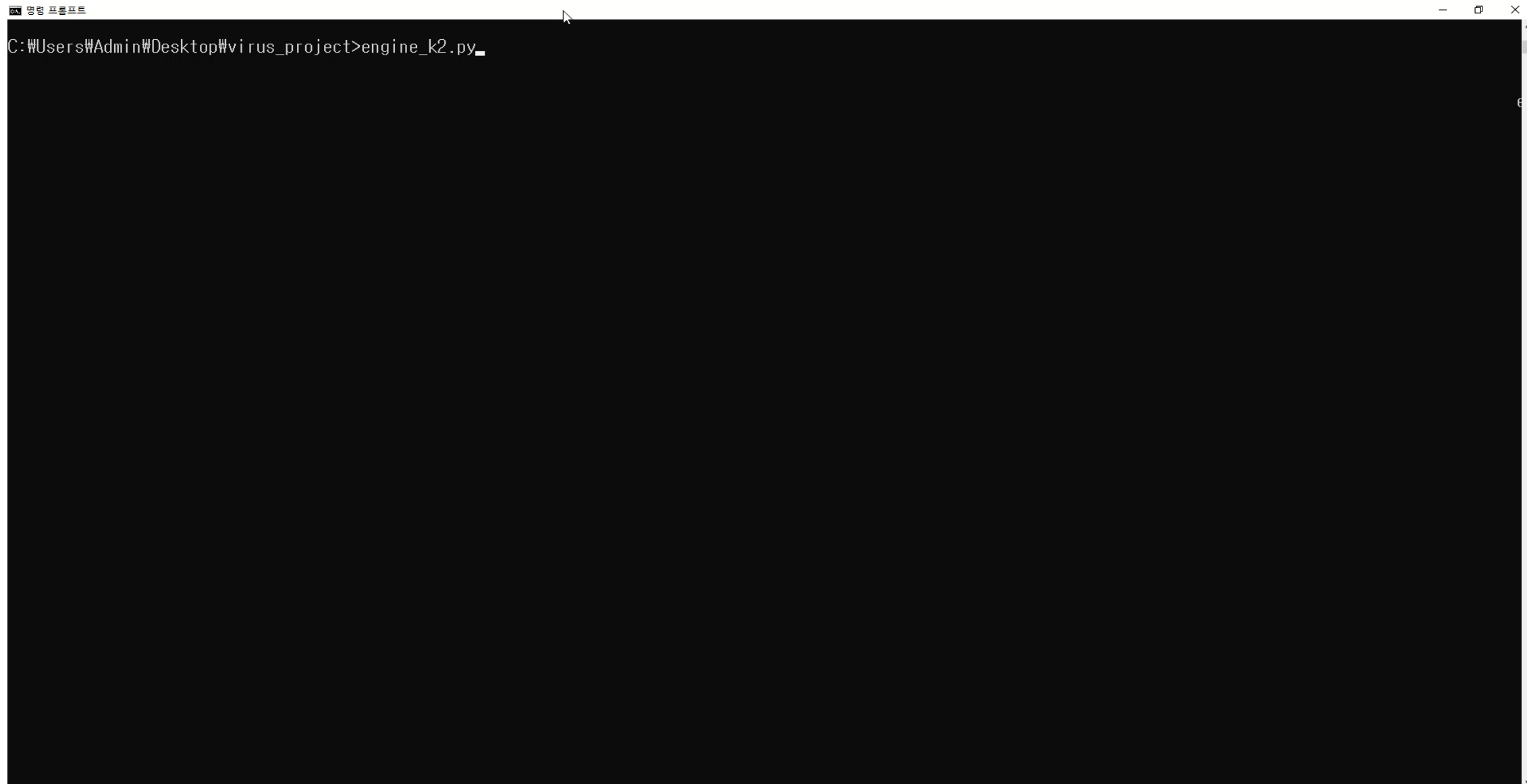
> 2022-12-23 @ 11:13:23.000 detection time: 2022-12-23 @ 11:13:23.000 field: virus file: malware_sample5.txt file path: C:\Users\Admin\Desktop\virus_project\virustest\malware_sample5.txt hostname: antipyrus
ip: 192.168.56.1 pc: DESKTOP-EF2BM5I username: user1 virus detection: malware_sample5 _id: yDvBPIUBrIUju-iNsIWx _index: antipy_log _score: - _type: _doc

> 2022-12-23 @ 11:13:22.000 detection time: 2022-12-23 @ 11:13:22.000 field: virus file: malware_sample3.txt file path: C:\Users\Admin\Desktop\virus_project\virustest\malware_sample3.txt hostname: antipyrus
ip: 192.168.56.1 pc: DESKTOP-EF2BM5I username: user1 virus detection: malware_sample3 _id: xjvBPIUBrIUju-iNqYXo _index: antipy_log _score: - _type: _doc
```

Dashboard 시각화



최종 시연



```
명령 프롬프트
C:\Users\Admin\Desktop\virus_project>engine_k2.py
```

오류 해결 과정

프로젝트를 진행하며 발생했던 오류들을 정리하였습니다.

활용방안

백신 프로그램을 활용할 수 있는 방법에 대해 생각해보았습니다.

프로젝트를 진행하며 얻은 점

프로젝트를 진행하면서 생긴 시행착오와 팀원들과 해결하는 과정을 통해 얻은 점을 정리하였습니다.

04

CONCLUSION

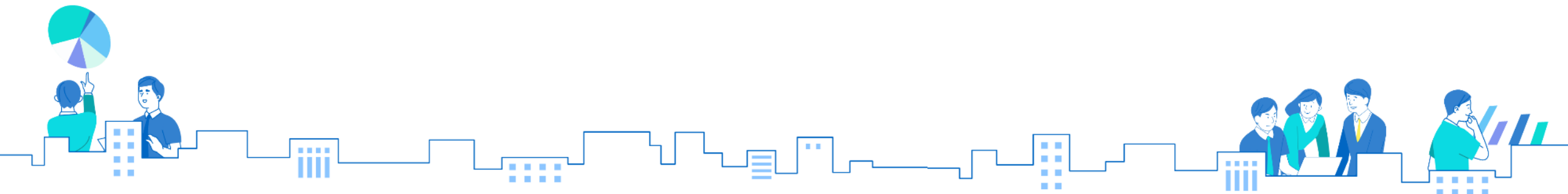


오류해결 과정

백신 개발 과정에서의 논리 오류 문제

문제	vname이 not defined로 표시
↓	
해결과정	vname변수 선언을 if문 밖에서 하도록 위치 수정
↓	
결과	vname이 실제 바이러스명으로 출력

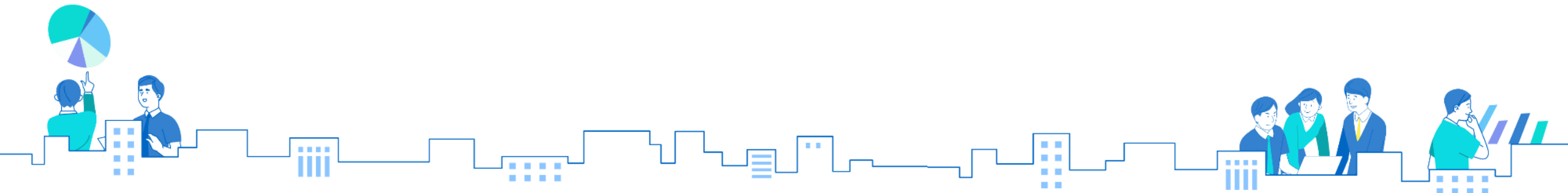
```
{'detection time': '2022-12-12 15:45:31', 'field': 'virus', 'file': 'malware1.txt', 'file path': 'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware1.txt', 'hostname': 'antipyrus', 'ip': '192.168.56.1', 'pc': 'DESKTOP-EF2BM51', 'username': 'user1', 'virus detection': 'malware1',  
b'C:\\Users\\Admin\\Desktop\\virus_project\\test\\malware1.txt'  
infected malware1
```



오류해결 과정

KIBANA 시각화 과정에서의 time zone 문제

문제	KIBANA의 기본 time zone은 browser로 자동 설정
↓	
해결과정	로그에서 datetime을 UTC로 표현하도록 값 변환
↓	
결과	서버 시간과 로그에서 찍히는 detection time 시간차 해결



활용방안



해당 백신의 mobile 버전을 만들어 노트북이 아닌 다른 환경에서도 백신을 사용하여 파일분석이 가능하도록 한다.

사용자가 백신을 직접 실행하지 않아도, 예약기능을 추가하여 원하는 시간에 맞게 자동으로 악성코드를 탐지할 수 있도록 한다.

검색과 시각화 기능을 통해서 최근에 탐지된 바이러스 종류와 침입횟수를 파악해 그에 맞는 보안솔루션까지 기대해 볼 수 있다.



프로젝트를 진행하며...



실무에서 개발자들이 협업을 위해 사용하는 깃허브를 실제 사용해 봄으로써 편리하고, 손쉽게 개발 과정을 공유할 수 있었다.



성과를 이뤄내는 과정을 팀원들과 함께 진행하면서 성취감과 동시에 개발 과정 중 협업이 얼마나 중요한 부분인지 느껴보는 경험이었다.



오픈소스를 직접 활용해 프로그램이 동작하면서 발생하는 로그를 수집하고 관제 업무분야에서 주로 사용하는 도구인 ELK를 통해 시각화 작업을 경험하였다.



다양한 명령어인 vi/yum/netstat/curl/sudo를 통해서 리눅스 환경을 구축해 보았다. 이를 통해 실무에서 많이 사용 되어지는 리눅스 환경에 익숙해지는 계기였다.



시각적 요소 추가

전체 프로젝트 진행 후 기능 상승을 위한 추가했으면 하는 부분입니다.

검사 가능한 파일 형식 추가

전체 프로젝트 진행 후 기능 상승을 위한 추가했으면 하는 부분입니다

Alerts

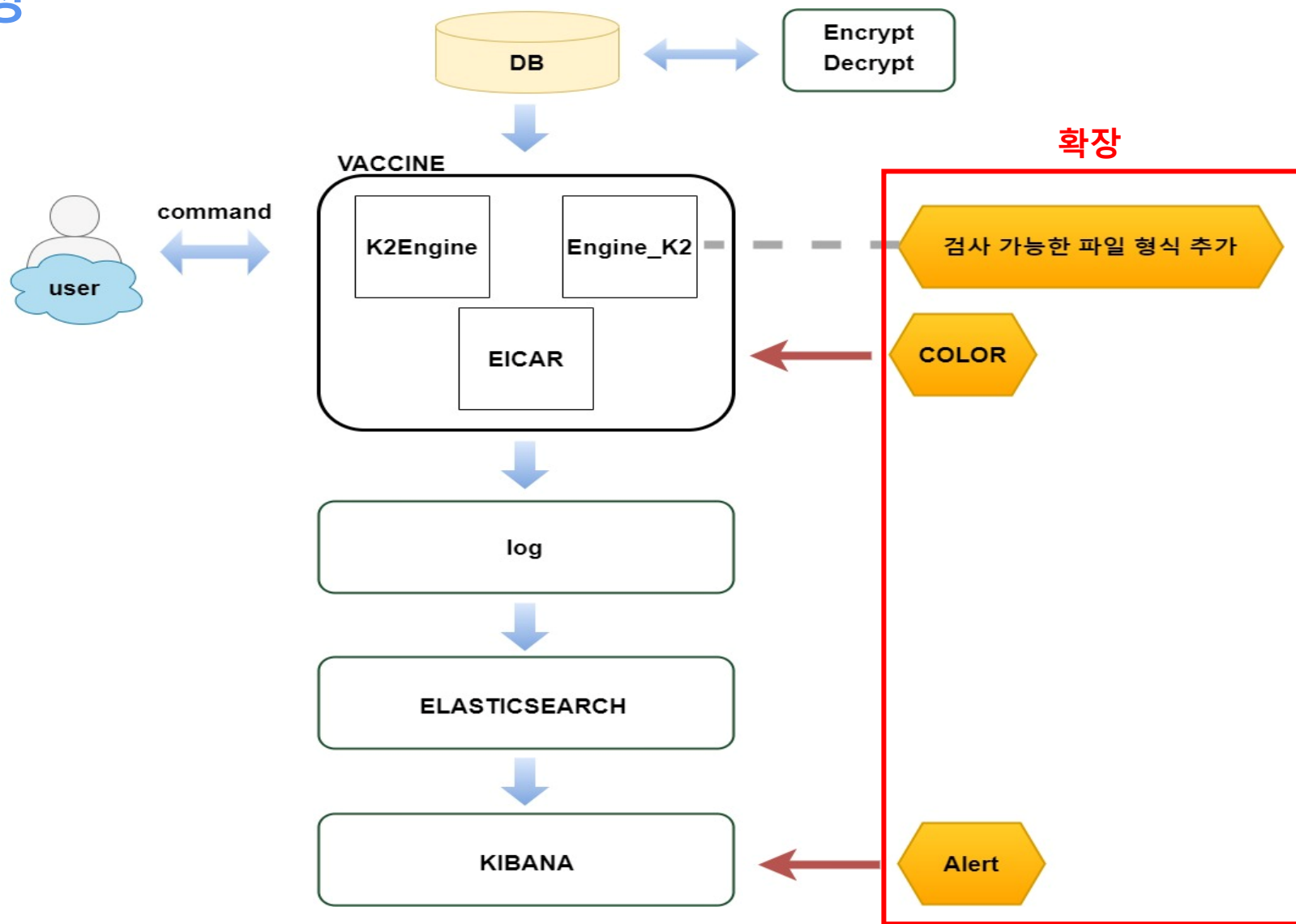
전체 프로젝트 진행 후 기능 상승을 위한 추가했으면 하는 부분입니다.

05

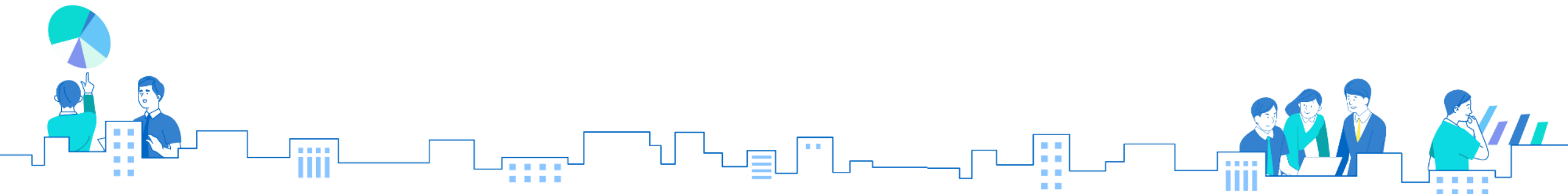
IMPROVEMENT



추가 개선 방향성



Q & A





THANK YOU

Anti_Pyrus