

10

10조(꿀벌🐝/허니팟 공격 분석 및 어택 라이브 맵 구축)

[주제] 허니팟 공격 분석 및 어택 라이브 맵 구축

저희 팀 프로젝트는 허니팟 구축을 통한 공격 분석 및 어택 라이브 맵 구축입니다.
사용자가 웹에 접속하면 대한민국을 공격하는 공격 패킷들을 실시간으로 볼 수 있으며
여러 분석 통계를 제공합니다.
또한, LLM과의 대화 및 머신러닝 분석 기능을 제공해 인사이트를 추가적으로 제공합니다.

1. 프로젝트 개요

프로젝트 필요성

📌 시민들의 보안의식 부족

<https://m.boannews.com/html/detail.html?idx=132077>

한국인들의 사이버 보안 인식이 2023년, 2024년 연속으로 최하위를 기록

실제로 얼마나 많은 공격들이 네트워크를 떠다니고 현재 우리나라, 그리고 자기자신을 공격하고 있는지 체감할 수 있게 하는 무언가가 필요하다고 느낌

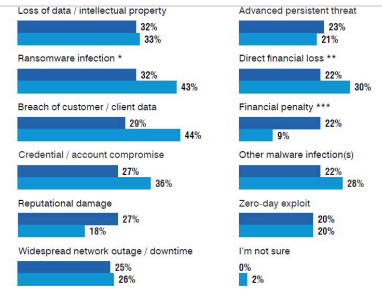
📌 구두 교육의 한계

"국내 기업 직원 62%, 보안 중요성 알지만 보안 해치는 행동은 지속"

프루프포인트(Proofpoint Inc.)가 올해로 10번째인 피싱 현황(State of the Phish) 보고서를 발표했다. 이 보고서에 따르면 기업 직원 62%가 랜섬웨어나 멀웨어 감염, 데이터 유출, 금전적 손실로 이어질 수 있는 피싱 리스크를 인지

<https://www.dailysecu.com/news/articleView.html?idxno=15391>

4



기업 조직 중 12%만이 TOAD 공격 인지·예방법을 교육하고 있는 것으로 나타났고, 생성형 AI 안전 교육을 실시하고 있는 조직도 24%에 그침

보안 공격의 74%는 여전히 인적 요인으로 인해 발생하며 보안 문화 조성은 중요하지만 일반적인 인식 교육만으로는 해결할 수 없는 문제가 존재

저희의 프로젝트를 통해 사람들은 우리의 웹사이트를 통해 실제 사이버상의 공격이 얼마나 많이 떠돌아다니고 우리가 얼마나 위협받고 있는지 체감하며 보안의식이 자연스럽게 올라갈거라 생각합니다.

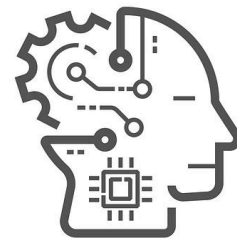
핵심 목표



어택 라이브 맵 제공



공격 세션들에 대한 다양한 통계를 제공



사용자와의 대화 및 상세 분석 기능 제공

사용자들이 눈으로 보고 **실제 체감**을 할 수 있도록 어택 라이브 맵을 이용하여 **실시간**으로 어떤 공격이, 어디서, 어떻게 들어오는지 정보 제공

분석 결과를 직관적으로 비교·시각화하여 일반인들도 쉽게 현재 들어오는 공격들이나 공격들의 통계를 쉽게 확인 가능

LLM 및 머신 러닝을 이용하여 모든 사용자들이 이용하기 편리하도록 챗봇 대화와 상세 분석 기능 지원

주요 기능 및 특징

1. 허니팟을 이용한 데이터 수집 및 처리

공격자 열어놓은 포트를 통해 접속이나 공격을 시도하면 로그를 수집하여 json으로 만들어 수집 및 처리.

2.어택 라이브 맵 생성 및 시각화

로그 JSON파일을 지도상에서 나타내고 로그들을 이용하여 통계 시각화.

3.사용자 친화적 로그 분석 서비스

LLM과 머신러닝을 이용한 대화와 추가 시각화 자료, 상세분석 등 사용자 친화적인 기능 제공.

할



팀 역

이름	역할	담당 업무
김민형	PM, 발표자	PM, 발표
이재홍	팀원	AWS EC2구축 및 S3 연동
임지혁	팀원	Aws EC2구축
최서희	팀원	웹,Imstudio
홍체영	팀원	웹

2. 사용 기술 및 개발 환경

사용한 기술

레이어	기술/도구	목적	비고
허니팟	Cowrie	인증 시도·명령·다운로드 시도 기록	배치 옵션: AWS EC2, Ubuntu 서버 (VM)
데이터 저장	AWS S3	전처리된 로그 저장	일정주기로 계속 업로드
웹 앱	Streamlit	라이브 공격 지도·대시보드	국가/도시 단위 핫스팟 및 타임라인 재생, Streamlit사용
시각화	Pydeck, Altair	지도·도넛/바/히트맵	툴팁 표준화
ML	Scikit-Learn	군집·이상치·패턴 설명	특징 추출 기반
챗봇	LM Studio 연계	질의응답/요약 생성	로그 기록을 LLM에 같이 전달

기능 요구 사항

1 로그 수집 및 처리

- 허니팟 서버구축 → 공격자가 허니팟 접속시 **cowrie**로 로그 수집
- 저장된 로그파싱 (세션별 분석→GeoLite2DB 이용 →IP를 위도,경도 등으로 변환 →Json처리)

2 로그 기반 라이브 맵 시각화

- 저장된 JSON을 기준으로 라이브맵 시각화 구현
- **pydeck**을 활용하여 실제 지도같이 표현

3 대시보드 분석

- 실시간 로그를 기반으로 한눈에 확인할 수 있는 **통계시각화**
ex)국가별 공격 발생 통계,최근 이벤트로그,공격유형 분포를 시각화 제공

4 LLM

- 수집된 로그 기록에 대한 궁금증을 자연어질문으로 입력
- 최신 로그데이터를 컨텍스트로 LLM에 전달→로그기반으로 답변 반환

5 머신러닝

- 비지도 학습을 통해 공격 로그들을 클러스터링
- 각 클러스터의 특징을 정리하여 시각화하여 표시

비기능 요구 사항

1 실시간 데이터 응답성

- 시스템은 실시간 이벤트 동기화 과정 중 데이터 지연이 상당하기에, 이 지연시간을 최소화해야함
- streamlit은 상호작용 발생시 전체 스크립트를 다시 실행하기 때문에 지연시간이 큼
- IP를 가지고 GeoLite DB를 조회할 때, 발생하는 지연시간이 큼

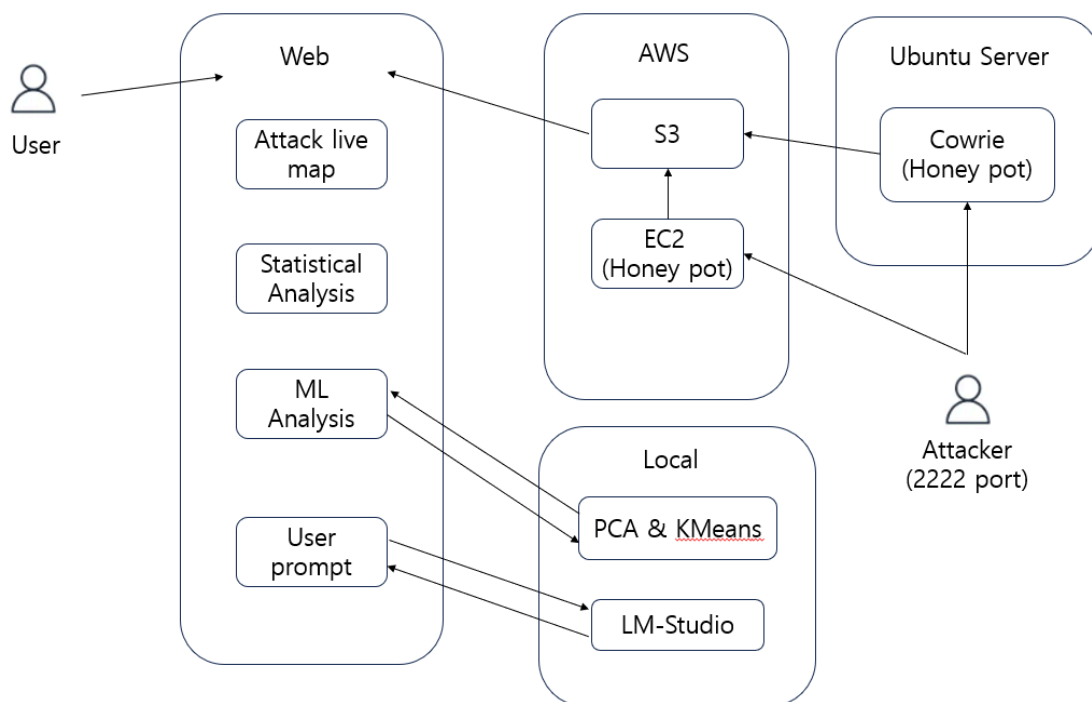
2 LLM속도 개선

- 공격 로그가 많아질시 분석/요약 지연 발생
- 공격 로그는 가능한 선(컴퓨팅 보유 수준)에서 건내주고 해당 로그를 통해 사용자에게 답변

3 사용자 친화 UI

- 사용자가 처음 사이트에 접속할 때, 눈에 확들어도록 UI 설계
- 여러가지 통계에 대해 그래프로 시각화하여 한눈에 들어올 수 있도록 컴포넌트 배치
- 사용자 응답대기시간 최소화

3.시스템 동작 흐름



1. 공격 로그 수집

- 공격자가 포트 스캔을 통해 2222번으로 공격하여 미끼서버(허니팟)으로 접속
- 핸드셰이크 및 세션 기록이 Cowrie를 통해 로그로 남음

2. 로그 전처리 및 위치 변환

1) 주기적 로그 갱신

- 파이썬 스크립트를 통해 cowrie.json 파일을 분석하고 재구성하여 events.json으로 만듭니다.
- 이 events.json은 주기적으로 S3에 계속 갱신되서 업로드됩니다.

2) GeoLite2로 src_ip → 위도/경도/국가 코드 변환

- GeoLite DB는 IP주소를 기반으로 지리적 위치정보를 조회해주는 DB입니다.
- 이 DB를 통해 공격자 IP로부터 공격자가 위치한 위도, 경도, 나라 정보를 얻게 됩니다.

3. Streamlit을 이용해 데이터 시각화 및 대시보드

- 최종적으로 만들어진 events.json의 정보를 통해 Streamlit 웹 페이지 구성
- 실시간 공격 맵: Pydeck ArcLayer & ScatterplotLayer로 Source→Destination 경로 표시
- 대시보드 통계: Altair그래프 및 카드형 UI로 국가별 공격통계, 공격 유형, 최근이벤트 제공
- LLM 및 머신러닝 : 데이터를 기반으로 AI기반 분석 제공

4. 주요 기능 상세 설명

허니팟 설정

허니팟 환경 구축

- AWS EC2에서 보안정책으로 인바운드 2222번 포트를 개방, 이후 Docker 환경에 Cowrie를 설치하여 허니팟 운영

- VM Ubuntu Server에 Cowrie 소스코드를 깃허브에서 다운받은 후, 와이파이 포트 포워딩을 설정

▼ 클라우드 인프라 설정

aws에서 ec2 보안그룹에서 2222포트를 열어놓고 생성

ec2 → s3 연결을 위해서

iam에서 testUser사용자 생성 후 s3 권한 부여 후

액세스 키를 이용하여 접근 예정

testUser
정보
삭제

요약

ARN

arn:aws:iam::913402647712:user/testUser

생성됨

August 20, 2025, 13:04 (UTC+09:00)

콘솔 액세스 비활성화됨

마지막 콘솔 로그인

-

액세스 키 1

Active

오늘 사용됨. 오늘 생성됨.

액세스 키 2

[액세스 키 만들기](#)

권한

그룹

태그 (1)

보안 자격 증명

마지막 액세스

권한 정책 (2)

삭제
[권한 추가 ▼](#)

사용자에게 직접 연결된 정책을 통해 또는 그룹을 통해 권한을 정의합니다.

필터링 기준 유형

검색

모든 유형 ▼

<

1

>

<input type="checkbox"/>	정책 이름	유형 ▼	연결 방식:
<input type="checkbox"/>	AmazonS3FullAccess	AWS 관리형	직접
<input type="checkbox"/>	CloudFrontFullAccess	AWS 관리형	직접

s3 버킷을 생성

10조(꿀벌🐝)/허니팟 공격 분석 및 어택 라이브 맵 구축)

7

범용 버킷 (1) 정보

[ARN 복사](#)[비어 있음](#)[삭제](#)[버킷 만들기](#)

버킷은 S3에 저장되는 데이터의 컨테이너입니다.

< 1 >

이름	AWS 리전	생성 날짜
<input type="radio"/> honeypot-bk	아시아 태평양(서울) ap-northeast-2	2025. 8. 19. pm 5:23:49 PM KST

ec2에서 s3로 파일을 보내기 위해서 role설정

권한 정책 (2) 정보

[시뮬레이션](#)[삭제](#)[권한 추가](#)

최대 10개의 관리형 정책을 연결할 수 있습니다.

필터링 기준 유형

모든 유형

< 1 >

정책 이름	유형	연결된 엔터티
<input type="checkbox"/> AmazonS3FullAccess	AWS 관리형	2
<input type="checkbox"/> CloudFrontFullAccess	AWS 관리형	3

임시로 모든 권한을 부여 추후 수정 예정

아래 ec2와 vm에서 보낸 파일모두 잘 들어옴을 확인.

< 1 > {

이름	유형	마지막 수정	크기	스토리지 클래스
<input type="checkbox"/> cowrie.log	log	2025. 8. 20. pm 4:43:03 PM KST	6.0MB	Standard
<input type="checkbox"/> events_rich.json	json	2025. 8. 20. am 9:25:13 AM KST	4.6KB	Standard
<input type="checkbox"/> events.json	json	2025. 8. 20. pm 4:43:03 PM KST	508.4KB	Standard

▼ VM 인프라 설정

Ubuntu Server 22.04 버전 VMWare에 설치 후 허니팟 설정


```

cowrie@aram:~/cowrie$ ./bin/cowrie start
Using default Python virtual environment "/home/cowrie/cowrie-env"
Starting cowrie: [twistd --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.1
ogger cowrie ]...
/home/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:110: Cryptograph
yDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.Tri
pleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b'3des-cbc': (algorithms.TripleDES, 24, modes.CBC),
/home/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:117: Cryptograph
yDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.Tri
pleDES and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.

```

와이파이 포트포워딩 설정을 통해 외부 22번(ssh)로 오는 모든 공격을 2222번 내부포트로 향하게 설정

포트 포워딩 설정

소스 IP 주소			
소스 포트		~	
외부 포트		~	
내부 IP 주소			
내부 포트		~	
프로토콜	TCP ▼		
설명			

추가

선택	소스 IP 주소	소스포트	외부포트	내부 IP 주소	내부 포트	프로토콜	설명	플래그
<input type="checkbox"/>			22	172.30.1.55	2222	TCP	honeypot	KT

이후 aws cli 설치 작업 진행 후 cowrie 로그를 s3에 옮기는 작업 진행 (이 작업에서 오랜 시간 소요)

공격 로그 파싱 & 레이블 추가 ([parser.py](#))

- cowrie.json에서 세션별 분석을 통한 label 생성

▼ 로그 파싱 및 레이블 생성

```

import json
import uuid
from collections import defaultdict

def get_geoiip_data(ip_address):
    # VMWare에 geolite.mmdb를 넣는데 한세월 걸려서 패스
    # 로컬에서 넣을 것
    return {"lat": None, "lon": None, "country": "N/A"}

```

```

def analyze_session(session_logs):
    """
    각 세션을 분석해서 label과 심각도를 내보냅니다.
    """
    # 기본 label은 단순 scan
    label = "scan"
    severity = 1

    commands = []
    has_successful_login = False
    has_failed_login = False

    for log in session_logs:
        event_id = log.get("eventid")
        if event_id == "cowrie.login.success":
            has_successful_login = True
        elif event_id == "cowrie.login.failed":
            has_failed_login = True
        elif event_id == "cowrie.command.input":
            commands.append(log.get("input", "").lower())

    if has_successful_login:
        # 로그인을 시도했고 성공했다면 reconnaissance(정찰)
        label = "reconnaissance"
        severity = 2
        # 명령어로 추가 구분
        for cmd in commands:
            if "miner" in cmd:
                return "cryptominer-check", 4 # High severity for cryptojacking
            attempts
            if "/ip cloud print" in cmd:
                return "mikrotik-recon", 3 # Specific recon for MikroTik routers
            if "telegramdesktop" in cmd or "smsd" in cmd:
                return "info-gathering", 3 # Attempt to steal user data
    elif has_failed_login:
        # 로그인 시도가 실패했다면 brute-force
        label = "brute-force"
        severity = 2

```

```
return label, severity
```

```
def parse_cowrie_logs(input_file="cowrie.json", output_file="events.json"):
```

```
    """
```

```
    Cowrie 로그 파일(JSON 형식)을 한 줄씩 읽어 처리한 후,  
    분석된 결과를 새로운 JSON 파일로 저장합니다.
```

```
    """
```

```
    sessions = defaultdict(list)
```

```
    # Step 1: session id를 기준으로 로그를 그룹화
```

```
    try:
```

```
        with open(input_file, 'r') as f:
```

```
            for line in f:
```

```
                try:
```

```
                    log = json.loads(line)
```

```
                    session_id = log.get("session")
```

```
                    if session_id:
```

```
                        sessions[session_id].append(log)
```

```
                except json.JSONDecodeError:
```

```
                    print(f"Warning: Skipping invalid JSON line: {line.strip()}")
```

```
                    continue
```

```
    except FileNotFoundError:
```

```
        print(f"Error: Input file '{input_file}' not found.")
```

```
    return
```

```
output_events = []
```

```
    # Step 2: 그룹화된 각 세션을 분석
```

```
    for session_id, logs in sessions.items():
```

```
        if not logs:
```

```
            continue
```

```
        # 로그를 시간순으로 정리
```

```
        logs.sort(key=lambda x: x.get("timestamp", ""))
```

```
        # 반드시 첫 이벤트는 connection 이벤트여야함
```

```
        first_log = logs[0]
```

```
        if first_log.get("eventid") != "cowrie.session.connect":
```

```

        continue # Skip sessions that don't start with a connection

# Step 3: 전체 세션을 위 함수로 분석
label, severity = analyze_session(logs)

src_ip = first_log.get("src_ip")

# geolite_data를 가져오는 부분은 패스
geoip_data = get_geoip_data(src_ip)

# Step 4: JSON 형식으로 데이터 생성
event = {
    "id": str(uuid.uuid4()),
    "ts": first_log.get("timestamp"),
    "src_ip": src_ip,
    "lat": geoip_data["lat"],
    "lon": geoip_data["lon"],
    "country": geoip_data["country"],
    "port": first_log.get("dst_port"),
    "proto": "tcp" if first_log.get("protocol") == "ssh" else first_log.get(
("protocol"),
    "label": label,
    "severity": severity
}
output_events.append(event)

# Step 5: events.json 작성
with open(output_file, 'w') as f:
    json.dump(output_events, f, indent=2)

print(f"✅ Success! Processed {len(sessions)} sessions and created '{output_file}'.")

if __name__ == "__main__":
    parse_cowrie_logs()

```

어떻게 각 세션 로그에 Label을 남겼는가?

1. 기본으로 label을 scan으로 설정 ⇒ 심각도 1

2. 로그인 시도로 구분

상황	레이블	심각도	의미
로그인 성공	reconnaissance	2	시스템에 침투 성공
로그인 실패	brute-force	2	무차별 대입 공격
연결 자체만 시도	scan	1	단순 포트 스캔

3. 성공한 로그인에 대해 시도한 명령어로 구분

명령어 패턴	레이블	심각도	공격 목적
miner	cryptominer-check	4	암호화폐 채굴 악성코드 설치
/ip cloud print	mikrotik-recon	3	MikroTik 라우터 정보 수집
telegramdesktop, smsd	info-gathering	3	개인정보 탈취 시도

- 이후 events.json으로 생성하여 s3에 보냄

GeoLite2-City.mmdb를 이용해 IP로부터 위도, 경도, 나라 데이터 반환 (`sync_daemon.py`)

- s3로부터 받은 json에는 ip 데이터 밖에 없음
- 다음과 같은 순서로 데이터 반환
 - events.json안의 ip를 추출
 - 해당 ip를 통해 GeoLite2 DB 조회
 - 해당 ip의 위도, 경도, 나라 데이터 반환
 - 기존 events.json에 위도, 경도, 나라를 추가하여 저장
- 주어진 주기에 맞춰 지속적으로 s3에 접근하여 events.json을 불러옴

Attack Live Map구현

- 저장된 로그를 기반으로 지도에 공격자의 위치 표시, 통계시각화

경로 코드 생성 (`1_🌐_실시간_공격_맵.py` , `utils.py`)

- Pydeck 라이브러리를 이용해 지도 시각화 → 선, 점, 궤적등의 레이어 얹기 가능
- ArcLayer/ScatterplotLayer
 - ArcLayer : 두 지점(공격자 → 허니팟 위치)을 곡선으로 연결[연결선]
 - ScatterplotLayer: 공격자의 위치를 점으로 표시[점]
- interp() : 선으로만 이으면 단순해보여 비행기처럼 날아가는 애니메이션 구현함수
 - ⇒ 좌표를 조금씩 나눠 찍어 점이 비행기처럼 날아가듯이 표현
 - ⇒ 공격자 위도와 허니팟의 위도의 거리사이의 중간 좌표를 20개쯤 생성한 애니메이션 추가

대시보드 통계 생성(`web/1_🌐_실시간_공격_맵.py` , `web/ui_components.py` ,

`data_handler.py` , `utils.py`)

- 데이터 처리
 - Cowrie로그를 DataFrame으로 변환 → 국가코드,국기컬럼 추가
 - `load_events()` 함수로 이벤트 정제
- 상단 메트릭 카드
 - `display_metrics(df)` 함수로 총 이벤트수,고유ip수,국가수, 마지막 갱신시간 표시
- 국가별 공격 랭킹
 - 간편 집계를 위한 Pandas,간편 통계를 내기 위한 Altair 이용
 - `df["country_code"].value_counts().head(10)` → 공격 top10 추출
 - `reset_index()` / `columns=[..., "count"]` → 시각화/표시용 컬럼 정리
 - `get_flag_emoji(code)` → 국가 코드 → 국기 이모지 변환 (파일: `utils.py`)

- Altair을 이용해 막대그래프 생성
⇒ 국가별 공격 상위10개를 국기이모지를 활용해 막대그래프로 표시됨
- 최근이벤트 테이블
 - 시간 순 정렬을 위해 **Pandas** 사용
 - `df.sort_values("ts", ascending=False).head(15)` → 최근 15건 추출
 - 표시 컬럼 예시: `["ts", "src_ip", "country_code", "label"]`
⇒ 최신 공격 흐름을 테이블로 빠르게 확인가능하게함 (시간/IP/국가/유형)
- 공격유형분포
 - 간편하게 차트 생성하기 위한 Altair 사용
 - `df["label"].value_counts().head(5)` → 공격 유형별 집계해서 Top5만 추출
 - `mark_arc(innerRadius=60)` → 파이차트 대신 **도넛 차트**로 표시
 - `tooltip=["label", "count"]` → 마우스 오버 시 **공격 유형·횟수** 표시
⇒ 상위 5개 공격 유형이 도넛차트로 표시

LM Studio 연동 및 머신러닝 분석

LM-Studio를 이용해 챗봇 기능 (`web/pages/_LLM_분석가.py`)

- 시스템 프롬프트와 최근 이벤트 5개를 함께 사용하여 LLM에게 전달
- 사용자 질문 + 시스템 프롬프트 + 최근 이벤트 5개
- 대답을 한번에 받으려면 유저의 대기시간이 너무 길어져 스트림 방식으로 처리

PCA 및 KMeans를 이용한 클러스터링 분석 (`web/pages/_AI_공격_분석서비스.py`)

- 머신러닝 분석을 위해 다음의 5개의 피쳐 선정 ⇒ 위도, 경도, 심각도, 레이블 인코딩이된 공격 유형, 공격한 "시간"
- 5개의 피쳐를 통해 S3에서 가져온 데이터를 가지고 다음의 순서대로 머신러닝 진행

1. KMeans에 사용할 최적의 K를 구하기 위해 엘보우 메소드 실행 (K를 1~10으로 설정해가며 Inertia 계산)
2. 최적의 K를 가지고 KMeans 클러스터링 진행 (비지도 학습)
3. 이후 하나의 그래프로 산점도를 표현하기 위해 주성분 분석(PCA) 사용
4. PCA를 통해 나온 2차원 좌표를 가지고 산점도 그래프 그리기
5. KMeans로 나온 각 클러스터의 국가 및 공격 유형 통계 보여주기

5. 메인 페이지 시각화 목록

1. 어택 라이브 맵

최근 들어온 로그 데이터에서 공격시도 횟수, 공격을 시도한 국가 수 등 전체적인 데이터들과 지도에 공격한 지역의 위치를 지도에서 표현하여 얼마나 공격이 들어오는지 체감할 수 있게 시각화하여 제공

전체적인 데이터를 일반적인 시각화로 보여주기보단 시도 횟수와 실제 공격받는 듯한 애니메이션으로 체감할 수 있도록 도와줌

2. 최근 이벤트 조회

공격 로그 데이터를 이용하여 공격의 전체적인 내용을 파악할 수 있게 사용자들이 익숙한 엑셀 형식으로 제공하여 언제, 어디서, 어떤 공격을 시도했는지 알 수 있도록 제공

사용자가 최근 이벤트와 다른 시각 자료들을 함께 보면 더욱 효과적으로 이해할 수 있음

3. 공격 국가 및 유형 랭킹

최근 들어온 로그 데이터 전체에서 아이피주소를 이용하여 위치를 파악 후 위치에 해당하는 각 나라별로 통계를 해서 공격시도 상위 10개 나라 공개하고, 어떤 공격시도를 했는지 사용자가 한 눈에 파악할 수 있도록 원형 차트 제공

어떤 국가에서 공격이 많이 들어오는지, 어떤 유형의 공격이 많이 들어오는지 눈으로 쉽게 확인 가능

6. AI 공격 분석 서비스 페이지 시각화 목록

1. PCA 산점도

5개의 특징을 2개의 특징으로 압축하기 위해 PCA를 사용. 이후 나온 2개의 좌표를 통해 각 공격 로그들을 2차원 좌표에 표시, 이때 같은 클러스터는 같은 색으로 표시

전체적인 데이터를 2차원 데이터로 압축하여 보여주기 때문에 가시성이 좋고 특징을 파악할 수 있음

2. 클러스터별 통계

KMeans를 통해 나온 각 클러스터의 국가 수 통계, 공격 유형 통계를 시각화하여 제공

각 클러스터의 특징을 한눈에 이해할 수 있음

7. LLM 분석가 페이지 시각화 목록

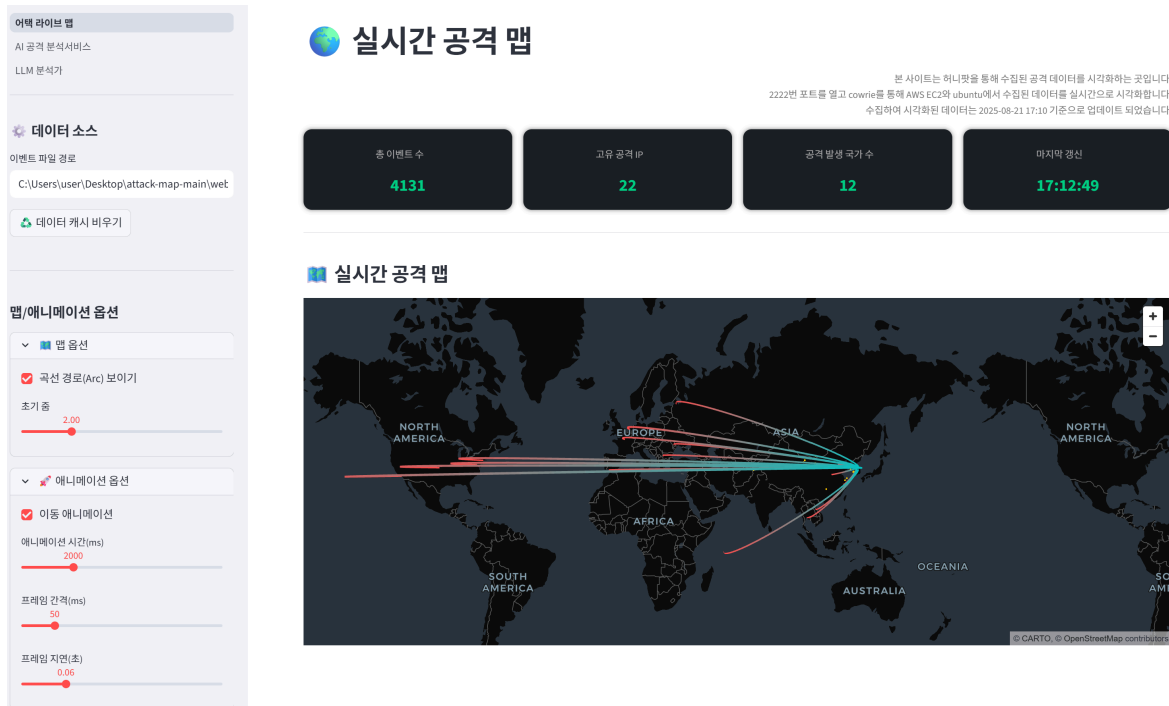
1. Chatbot 인터페이스

최근 들어온 로그 데이터를 가지고 사용자와 얘기 할 수 있는 채팅 인터페이스 표시

LLM과의 대화를 통해 보다 웹사이트와 사이트가 상호작용하는 느낌을 줌

8. UI 및 결과 화면

[Page 1] 메인페이지



- 사용자는 육안으로 쉽게 공격현황을 확인 할 수 있음 [빨간색(공격자)→ 푸른색(허니 팻)]

사이드바 메뉴 제공

- [어택 라이브맵]
- [AI 공격 분석 서비스]
- [LLM 분석가]
- 사이드바에 애니메이션 효과를 ON/OFF설정 지원

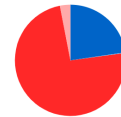
[Page 1] 공격로그 통계화면



최근 이벤트

ts	src_ip	country	label
2025-09-20 17:26:38+00:00	51.139.14.251	FR	scan
2025-09-20 17:26:38+00:00	20.189.105.85	US	scan
2025-09-20 17:26:37+00:00	20.189.105.85	US	scan
2025-09-20 16:41:24+00:00	198.235.24.224	US	scan
2025-09-20 16:41:24+00:00	45.140.17.26	RU	scan
2025-09-20 16:40:58+00:00	45.140.17.26	RU	scan
2025-09-20 16:08:49+00:00	198.235.24.224	US	brute-force
2025-09-20 16:08:49+00:00	198.235.24.224	US	brute-force
2025-09-20 16:08:49+00:00	198.235.24.224	US	reconnaissance
2025-09-20 16:07:47+00:00	198.235.24.224	US	brute-force

공격 유형 행렬 (Top 5)



Attack Line Map - Inspired by Shodan.io
Built with using Mapbox, Pydeck & Altair

- 공격당한 로그를 기반으로 쉽게 통계를 확인할 수 있음

[Page 2] LM Studio챗봇

LLM 기반 로그 분석가

최근 공격 이벤트 5개를 기반으로 상황을 요약하거나 **공격원 정보**를 질문할 수 있습니다.

예: 안녕하세요! 저는 최신 보안 로그를 분석하는 AI입니다. 무엇을 도와드릴까요?

질문을 입력하세요...

예: 제일 공격이 많이 들어온 유형은 뭐야?

- 로그 분석 결과를 요약하면 다음과 같습니다.
- 공격이 가장 많이 들어온 유형은 TCP 스캔입니다.
 - 총 5개의 로그 중 4개가 TCP 스캔으로 인한 공격입니다.
 - 공격이 들어온 국가로는 미국(US)과 프랑스(FR), 러시아(RU)가 있습니다.

분석 컨텍스트

이는 아래의 최신 이벤트 5개를 기반으로 답변합니다.

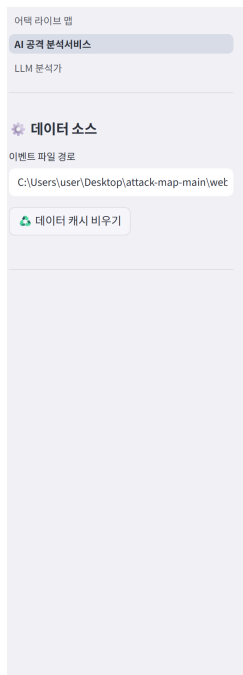
ts	src_ip	country_code	label
2025-09-20 17:26:38+00:00	51.139.14.251	FR	scan
2025-09-20 17:26:38+00:00	20.189.105.85	US	scan
2025-09-20 17:26:37+00:00	20.189.105.85	US	scan
2025-09-20 16:41:24+00:00	198.235.24.224	US	scan
2025-09-20 16:41:24+00:00	45.140.17.26	RU	scan

질문 해서 보기

Attack Line Map - Inspired by Shodan.io
Built with using Mapbox, Pydeck & Altair

- 공격로그를 기반으로 사용자의 궁금증을 해결하기위한 챗봇을 이용할 수 있음

[Page 3] PCA분석 ML

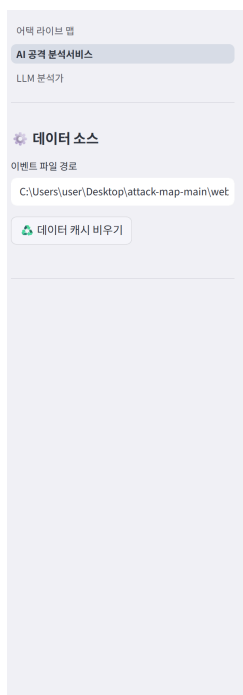
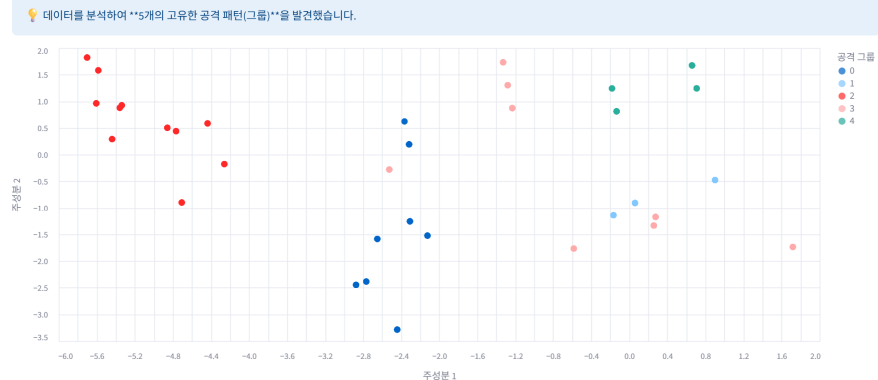


머신러닝 기반 공격 분석

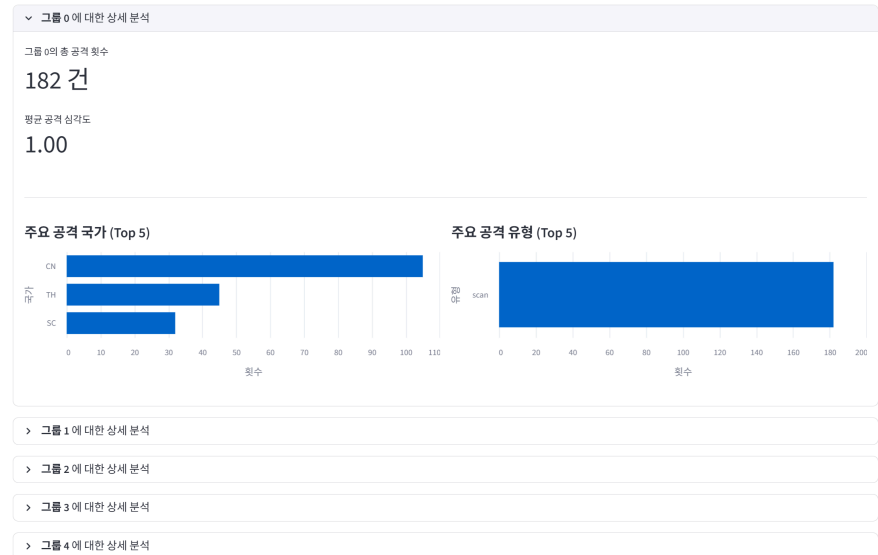
KMeans를 사용하여 모델이 공격 데이터의 특징을 스스로 학습하여 클러스터링을 진행합니다. 여기서 사용하는 특징은 위도, 경도, 심각도, 시간, 공격유형의 5가지입니다.

이후 PCA를 통해 5가지 특징을 2차원으로 압축하여 주성분 축1, 주성분 축2를 기준으로 시각화하여 표시합니다. 산점도 그림 아래에 각 클러스터별 특징이 정리되어 있습니다.

PCA 분석 결과



그룹별 공격 패턴 프로파일링



Attack Live Map • Inspired by Shodan.io
Built with using Streamlit, Pydeck & Altair

- 머신러닝 기반 PCA 분석 및 그룹 별 상세 분석을 확인할 수 있음.

9. 시연 영상

✓ 시연 영상

attachment:23435b32-6cb8-46b6-b947-34271050f808:10조_시연영상_완전
최종.mp4

10. 개발 진행 과정

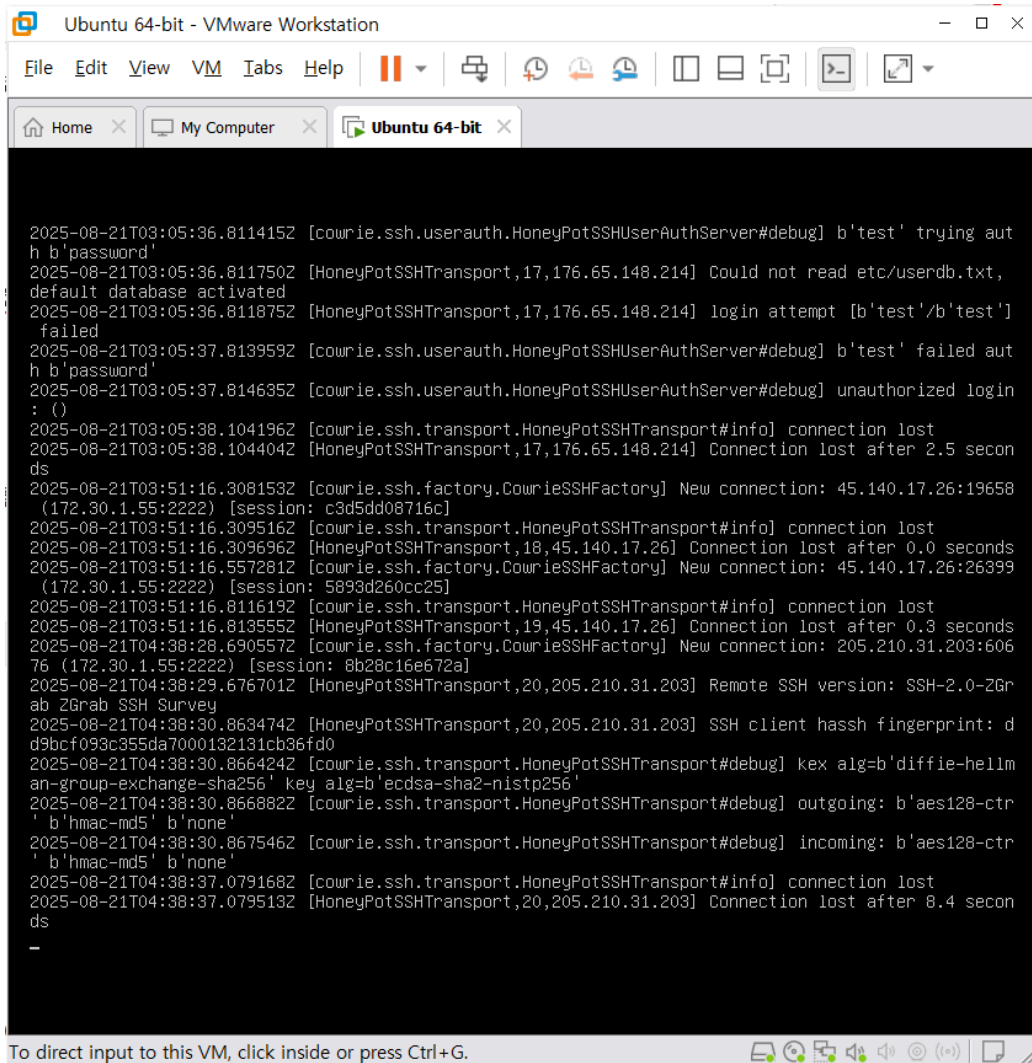
어려웠던 점과 해결 과정

문제점 1: 공격 데이터의 부족

- AWS EC2에서 포트를 열고 Cowrie를 실행하여 허니팟 운영을 해보았지만 들어오는 공격의 수가 너무 적었음.
- 데이터가 너무 적기에 통계를 내거나 AI 학습을 진행하기에 너무 어려웠던 상황
- 그렇다고 타 데이터 셋을 그냥 사용해버리면 "라이브"의 의미가 아예 사라져버림.

해결 과정

✅ Ubuntu Server에 허니팟을 추가로 설치하여 해결



```
2025-08-21T03:05:36.811415Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'test' trying aut
h b'password'
2025-08-21T03:05:36.811750Z [HoneyPotSSHTransport,17,176.65.148.214] Could not read etc/userdb.txt,
default database activated
2025-08-21T03:05:36.811875Z [HoneyPotSSHTransport,17,176.65.148.214] login attempt [b'test'/b'test']
failed
2025-08-21T03:05:37.813959Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'test' failed aut
h b'password'
2025-08-21T03:05:37.814635Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login
: ()
2025-08-21T03:05:38.104196Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-08-21T03:05:38.104404Z [HoneyPotSSHTransport,17,176.65.148.214] Connection lost after 2.5 secon
ds
2025-08-21T03:51:16.308153Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 45.140.17.26:19658
(172.30.1.55:2222) [session: c3d5dd08716c]
2025-08-21T03:51:16.309516Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-08-21T03:51:16.309696Z [HoneyPotSSHTransport,18,45.140.17.26] Connection lost after 0.0 seconds
2025-08-21T03:51:16.557281Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 45.140.17.26:26399
(172.30.1.55:2222) [session: 5893d260cc25]
2025-08-21T03:51:16.811619Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-08-21T03:51:16.813555Z [HoneyPotSSHTransport,19,45.140.17.26] Connection lost after 0.3 seconds
2025-08-21T04:38:28.690557Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 205.210.31.203:606
76 (172.30.1.55:2222) [session: 8b28c16e672a]
2025-08-21T04:38:29.676701Z [HoneyPotSSHTransport,20,205.210.31.203] Remote SSH version: SSH-2.0-ZGr
ab 2GGrab SSH Survey
2025-08-21T04:38:30.863474Z [HoneyPotSSHTransport,20,205.210.31.203] SSH client hassh fingerprint: d
d9bcf093c355da7000132131cb36fd0
2025-08-21T04:38:30.866424Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] kex alg=b'diffie-hellm
an-group-exchange-sha256' key alg=b'ecdsa-sha2-nistp256'
2025-08-21T04:38:30.866882Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] outgoing: b'aes128-ctr
' b'hmac-md5' b'none'
2025-08-21T04:38:30.867546Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes128-ctr
' b'hmac-md5' b'none'
2025-08-21T04:38:37.079168Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-08-21T04:38:37.079513Z [HoneyPotSSHTransport,20,205.210.31.203] Connection lost after 8.4 secon
ds
-
```

- 와이파이 포트 포워딩 설정을 통해 Local 컴퓨터 내부 VM에 허니팟 설치.
- EC2에 비해 많은 공격이 확보되었음.
- AWS EC2의 공격로그와 VM의 공격로그를 통합하여 진행.

문제점 2: GeoLiteDB 조회 딜레이

- 모든 데이터를 하나씩 확인하며 IP 조회를 위해 GeoLiteDB에 접근하여 동일 IP가 계속나와도 반복하여 딜레이가 심했음.
- 공격 로그들을 분석해보며 같은 IP가 여러번 공격을 하는 경우가 많기에 이는 매우 비효율적임.

해결 과정

- ✓ IP 조회결과를 캐싱함으로써 해결

- 다음과 같은 순서로 문제를 해결
1. 고유 IP 추출: 먼저 S3에서 받은 전체 데이터에서 위치 정보가 없는 IP들을 중복 없이 모두 추출.
 2. 이 고유한 IP 목록에 대해서만 GeoIP 조회를 수행하여 캐시에 미리 채움.
 3. 전체 이벤트 목록을 다시 돌면서, 이미 캐시된 위치 정보를 각 이벤트에 매우 빠르게 채움.

```
@lru_cache(maxsize=4096) # 최대 4096개의 IP 조회 결과를 메모리에 저장
def _get_location_for_ip(ip: str) → Optional[Dict[str, Any]]:
    ...
```

문제점 3: S3 버킷 연동

- AWS EC2와 VMware의 ubuntu 데이터들을 S3에 실시간으로 데이터를 올리려고 하는 과정에서
EC2와 S3 연결은 클라우드 수업시간에 배웠기 때문에 비교적 수월. 하지만 ubuntu 데이터를 올릴때 어려움 발생
- 또 한 S3 버킷에 올라간 데이터를 다시 실시간으로 불러와서 사용하는 코드를 작성할 때 어려움 발생

해결 과정

- ✅ S3 접근 권한을 가진 IAM 사용자 생성 후 액세스 키 발급

testUser
정보
삭제

요약

ARN

arn:aws:iam::913402647712:user/testUser

생성됨

August 20, 2025, 13:04 (UTC+09:00)

콘솔 액세스

비활성화됨

마지막 콘솔 로그인

-

액세스 키 1

Active

오늘 사용됨. 오늘 생성됨.

액세스 키 2

액세스 키 만들기

권한

그룹

태그 (1)

보안 자격 증명

마지막 액세스

권한 정책 (2)

사용자에게 직접 연결된 정책을 통해 또는 그룹을 통해 권한을 정의합니다.

필터링 기준 유형

모든 유형

검색

모든 유형

< 1 >

정책 이름

▲

유형

▼

연결 방식:

AmazonS3FullAccess

AWS 관리형

직접

CloudFrontFullAccess

AWS 관리형

직접

- 해당 액세스 키와 비밀 액세스키, 리전, 버킷이름을 이용하여 S3 버킷에 접근하여 로그 데이터인 events.json파일에 접근

```

AWS_ACCESS_KEY_ID=test
AWS_SECRET_ACCESS_KEY=tset
AWS_REGION=ap-northeast-2
S3_BUCKET_NAME=honey-pot-bk
S3_EVENTS_FILE_KEY=events.json

```

- 실시간으로 보내고 불러와야되기때문에 .env에 정리해놓고 사용

문제점 4: streamlit

- 생동감을 위해 애니메이션을 추가 했지만 끊김현상이 심했음→데이터수가 많아서

해결 과정

- 데이터 표시량 축소 → 최근 N개만 tail()
- 애니메이션 파라미터 보수적 운영

10조(꿀벌)/허니팟 공격 분석 및 어택 라이브 맵 구축

24

- `duration_ms` 짧게 (2초 수준)
- `frame_step` 크게 (프레임 수 줄임)

⇒ 꿈김현상이 많이 나아짐

11. 개선 사항 및 향후 계획

향후 계획

✅ AI를 이용한 보안 분석 고도화

- AI 모델을 활용한 공격패턴,방안법 설명기능 추가
- 이러한 허니팟 공격을 전문으로 AI를 파인튜닝하여 사용해보기

✅ UI개선

- 응답속도가 지금도 불만족스럽기 때문에 속도 개선 시도
- Streamlit이 아니라 React를 사용해서 좀 더 생동감 있는 UI 구현 시도

✅ 성능 최적화

- 대량 이벤트 발생시에도 원활한 환경 조성
- 지금은 2개의 허니팟이지만 10개 이상의 허니팟을 운영하며 다양한 로그 수집 시도 및 통합

✅ LLM 기능

- 로그를 분석해 대응방안까지 추천해주는 기능 추가
- 더 많은 데이터를 빠른 속도로 처리하여 응답할 수 있도록 로컬 성능 향상

12. 프로젝트 회고 및 소감

🐼 김민형

어디선가 보기만 하던 어택맵을 실제로 만들어보는 것이 굉장한 경험이 되었습니다. 실제 서버를 구축하고 그 서버에 오픈소스 허니팟을 이용해 실시간으로 공격 데이터를 받아보고, 그 데이터를 통해 무언가 분석을 하고 맵을 만드는 것이 재밌었던 것 같습니다. 여러가지 어려운 점들 많았고 힘들었지만 그만큼 값진 프로젝트가 된 것 같습니다. 팀원들 모두 고맙습니다~~!

이재홍

프로젝트를 진행하면서 수업 시간에 다뤘던 내용들이 생각보다 많아서 놀랐고, 복습하듯 직접 다시 구현해 봄으로써 어느 정도 머릿속에서 정리된 것 같아 기분이 좋았습니다.

또 sk 실더스에서 봤던 SECUDIUM 같이 영화에서 볼 법한 실시간 공격 맵과 각종 통계들을 구현해 보고, 좋은 경험이었습니다! 다들 고생 많으셨고 수고하셨습니다!!

임지혁

웹 페이지를 구성하면서 추가하고 싶은 기능이 있었으나, 기능 추가에 어려움이 있어 추가하지 못하여 아쉬움이 있었고 이런 경험이 처음이라 어려웠지만 재미 있었고

좋은 팀원분들 덕분에 제가 할 수 없는 부분들을 많이 도움을 받아서 큰 어려움 없이 프로젝트를 수행할 수 있었습니다. 다들 너무 고맙고 수고하셨습니다😊

최서희

프로젝트하면서 이걸 할 수 있을까?라고 두려웠는데 팀원들과 같이 진행하다 보니 5일이 지나고 저희가 하고자하는 프로젝트도 완수한 것 같습니다. 수업시간에 배웠던 지식들이 이렇게 프로젝트에 활용할 수 있고 이런결과를 낼 수 있다는 것에 대해 신기했던 시간이었습니다.특히 지도상에 로그를 표시하고 시각화할때 매우 신기했습니다! 팀원분들도 고생많셨습니다...!

홍체영

클라우드 보안을 일주일간 배웠지만 큰 주제만 있고 작은주제를 팀원들끼리 정하는 이번 프로젝트를 통해 여러가지를 배우는 유익한 시간이 되었습니다


AWS S3와 연동하여 데이터를 안정적으로 동기화하는 기능 구현은 클라우드 환경에 대한 자신감을 키우는 계기가 되었습니다.


또 LM Studio를 통해 챗봇을 만들어 Streamlit과 연결하는과정도 배우고 팀원들과 이런 저런 의견을 내며 소통하는 시간이 너무 좋았습니다

팀원 분들 모두모두 고생하셨습니다!!!

▼ Old

1일차 — 아이디어 & 기획



 아이디어 회의 내용

 R & R


 기획안(초안)


-  Kaggle 데이터셋

협력툴

-  구글 드라이브
-  Discord 채널

2일차 — 기능 설계 & 세팅


 기능 정의서(초안) (1)

 허니팟 설정 (1)


 화면 (1)


3일차 — 개발 & 최적화

 개발 문서 (1).

 속도 개선 기록 (1).

4일차

 최종 발표 자료(초안) (1).

 Lesson-Learned 작성 : 프로젝트 회고(느낀점, 배운점, 오류 등) (1).

▼ 시연영상제작

attachment:d5f09668-d197-4560-8ab8-53acbf5f7ac7:10조_시연영
상_최종.mp4

5일차

발표!!!!!!!!!!!!!!!!!!!!!!!!!!!!

 **강사님 피드백**

강사님 피드백 (1).

<https://discord.com/channels/1406823376823128135/1406824940375900171>

<https://discord.com/channels/1406823376823128135/1406824940375900171>
