회의록

—------------------------------------------------------------------------------------------

회의 제목 : 프로젝트 진행 현황 보고

회의 일시 : 2025-10-19, 오후 2시

장소 : KakaoTalk

참석자 : 고재현, 심수민, 전영우

회의 목적 : 개발 1주차 진행 현황 체크, 문제점 파악 및 해결 방안 모색색

조원 Epic 별 개발 현황 체크

—------------------------------------------------------------------------------------------

**E-1 피싱/스미싱 탐지**

고재현 조원

**S1-1 사용자는 이메일 본문 내 URL이 피싱 의심일 경우 자동으로 탐지 및 표기되길 원한다**

**1. 진행도/구현내용**

**1.1 진행 완료된 사항 및 설명**

**Gmail API 연동 (OAuth 2.0)**

* Google Cloud Console에서 OAuth 2.0 클라이언트 ID 발급 완료
* Gmail API를 통한 메일 자동 수신 기능 구현 완료
* 메일 제목, 발신자, 본문, 첨부파일 정보 추출 가능
* REST API 엔드포인트 제공 (/api/gmail/fetch)

**Naver Mail API 연동 (IMAP)**

* Naver IMAP 프로토콜을 이용한 메일 수신 기능 구현
* 앱 비밀번호 방식으로 인증 (2차 인증 불필요)
* 최근 메일 목록 조회 및 상세 정보 가져오기 완료

**URL 추출 및 분류 알고리즘**

* 정규식을 활용한 이메일 본문 URL 자동 추출 기능
* 의심스러운 도메인 패턴 분류 (단축 URL, .xyz, .top 등)
* 추출된 URL 정보를 EmailDto에 저장**피싱 탐지 엔진 (룰 기반)**
* 한글/영문 피싱 의심 키워드 탐지 ("긴급", "계좌", "클릭", "urgent" 등)
* URL 패턴 기반 위험도 분석
* 발신자 위조 여부 체크 (금융기관, 정부기관 사칭)
* 위험 첨부파일 확장자 검사 (.exe, .bat, .zip 등)
* 0-100 점수 기반 위험도 평가 (SAFE/SUSPICIOUS/DANGEROUS)

**데이터 전송 객체 (DTO) 설계**

EmailDto 클래스 구현 완료

메일 기본 정보, URL 목록, 피싱 탐지 결과 포함

* Lombok을 활용한 코드 간소화

**1.2 진행 중인 부분**

**데이터베이스 연동 준비**

H2 Database 또는 MySQL 연동 검토 중

Entity 설계 및 Repository 구조 계획 중

메일 정보 및 URL 분류 결과 저장 기능 구현 예정

**application.properties 설정 파일 작성**

* 민감 정보 분리 (credentials.json, API 키 등)
* application.properties.example 파일로 예시 제공 계획

—---------------------------------------------------------------------------------------------------------

전영우 조원

**S1-2 시스템은 의심 URL 클릭시 접속을 차단하고 경고 메시지를 제공해야한다.**

MySQL DB 저장된 도메인 기반, 접속 리다이렉션 검사 & 차단 시스템 구현

\* 사용법 :

1) MySQL 에 ReadMe 파일 에 있는 쿼리 실행 (DB Table 생성)   
2) github jeonyeongwoo text 리포지토리 내 첨부 스프링부트 파일 다운 및 실행   
3) cmd 에서 다음 문장 입력   
curl -X POST http://localhost:8080/api/admin/import/all   
4) http://localhost:8080/api/url/check?url=https://{url} << url 위치에 접속 원하는 url 입력 시 차단알림 표시   
   
해결 과제 : T2-1 / T2-3 / T2-4 / T2-5   
미해결 과제 : T2-2 (백엔드 내에서 html 형식 내보내는 형식 출력. 프론트 형태의 처리 x)   
   
협의 필요 내용 :

추후 통합 시 문제 생기지 않도록 소통 원활히 하도록 하자.

백엔드 처리 로직 -> 필요에 따른 조정

S1-3

업로드한 "제목, 본문 내용 csv 파일" 기반   
테스팅, 예측 데이터셋 분류 및 예측 하는 코드 생성   
   
이용 방법 : 1) 코드 colab 환경에 업로드 -> 블록 차례대로 실행 (목, 본문, 라벨 기반 csv파일 업로드 후 -> 아래의 블록 차례대로 실행하면 예시 샘플에 대한 예측 결과 출력)   
   
해결해야 할 과제 :   
1) 한국어 데이터셋 확보 <- 악의적 용도 활용 가능성 이유로 풀려있지 않음.   
2) 모델 정확도 향상 필요. <- gpt 임시 데이터셋 이유인지/ 혹은 모델 문제점인지 파악 및 개선 필요함.   
3) llm <- 을 어떻게 프로젝트에 연동시킬 수 있는지 방법 찾을 필요 있음.

해결 과제 :   
미해결 과제 : T3-1, T3-2, T3-3, T3-4

**해결 방안 탐색**

**문제 - LLM 관련 API 탐색**

1) openAI API 기반, 텍스트 임베딩 을 활용한 예측 및 활용   
https://wikidocs.net/200466

설명 : openAI API 에서 프롬프트 입력 시, 수치를 내뱉음 -> 이를 이용해서 텍스트 간 유사도 측정 위해 사용할 수 있을 것 같음.

장 / 단점 분석 :

장점 - 텍스트 임베딩을 이용해서 내보내는 수치 간 유사성 이용해 악성 메일 / 정상 메일 분류에 사용할 수 있을 것 같음   
 단점 - openAI API 사용하기 위한 비용 발생 5$ per 1 million token

**문제 - 한국어 데이터셋 확보**

1) 직접 메일 데이터셋 만든다.

직접 개인 메일 탐색해서 학습시킬 데이터셋을 만드는 방법

장 / 단점 분석 :

장점 - 타 기관에 찾아가거나 / 문의할 필요 없음.   
 단점 - 직접 데이터셋 만드는 시간 투자 필요 / 메일 분류에 인간 주관이 개입됨.

데이터셋 생성 과정에서의 개인정보 유출

2) 일단 영어 데이터셋 사용한다.

장점 - 타 기관 찾아가거나 문의 필요 X

직접 데이터셋 만들 필요 없음 / 데이터셋 생성 과정 개인정보 유출 위험 X

단점 - 데이터셋 판독 필요.

원래 취지와 달라짐 - 한글 메일 대한 것인데, 영어로 학습

데이터셋 :<https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/code>

※ 의문 : 스팸 / 정상 메일 데이터셋 : 직접 만들거나 혹은 영어로 우선 대체 가능하지만.

**스팸 메일 == 악성 메일 이라고는 할 수 없지 않나요?** 라는 의문이 있습니다.

—---------------------------------------------------------------------------------------------------------

심수민 조원님

**E-2 피싱/스미싱 탐지**

1. **진행도/구현내용**
2. **진행 완료된 사항 및 설명.**

* 서비스에서 지원할 이메일(Gmail, Naver mail)에 대해 이메일 정보를 가져올 수 있는지, 어떤 형태로 가져올 수 있는지 테스트 완료.

* VirusTotal API 연동 및 JSON 데이터 파싱
* 판별에 필요한 정보 가져오는 기능은 구현됨.
* 현재 테스트 페이지에서는 파일에 대한 JSON 형태의 데이터를 출력하고 있으며, UI 구현 후 연동할 예정.

* DB(mysql) 연동 및 첨부파일 데이터 저장
* 첨부파일 hash 값에 대한 정보를 데이터베이스에 저장하는 기능 구현됨.
* 추후 메일 분석 결과 최종 보고서를 보여주는 기능 구현 시 사용될 데이터.

1. **진행 중인 부분**

* 현재 가져온 데이터로 위험도 출력 및 경고 메시지 출력하도록 코드 작성 중.

1. **진행 예정**

* 첨부파일 종류별 구분 기능
* 압축 파일의 경우 별도의 압축해제 과정이 필요하므로 분기 작성 예정.
* 그외에 문서 파일/실행 파일은 파일명 또는 매직넘버(시그니처)로 구분할 예정.
* hash 연산 기능을 구현해 앞서 구현한 기능들과 연동할 예정.

1. **막히는 부분/협의해야 할 사항**

* 첨부파일 검사를 어느 시점에서 할 것인지, 고민중
* 사용자가 선택한 메일, 실시간으로 메일 검사 진행하는 기능 두가지가 있음.
* 실시간 메일 검사 -> 메일 받아오는 기능 구현 방식에 따라 달라질 것 같다?
* 사용자가 선택한 메일에 대해 검사하도록 우선 구현해야 할 것 같다.
* <https://github.com/Chungbuk-Genesis/MailGuard/issues/1>
* 학습 데이터셋을 직접 생성할 수 있다면 해야할 듯하다.
* 한국어로 된 메일 데이터셋을 구하기 힘들다.

* 주요 DB를 어떻게 설계할지 팀원들과 이야기해봐야 할 것 같다.
* 구체적으로 전부 설계하진 않더라도, 추후 통합 과정에서 문제가 발생하지 않도록 상의는 필요할 것 같다.

* 개발 환경 공유 필요.
* 각자 개발하는 IDE나 JDK 버전 등은 공유를 해야 할 것 같다.

1. **문제점**

* VirusTotal API 사용 자체에는 문제가 없으나, 일반 사용자의 경우 limit가 존재한다.
* 분당 4번 요청 가능/하루 500번 요청 가능
* 테스트 및 개발에는 문제가 없을 것 같지만, 실제 서비스로 사용되려면 일반 사용자로 개발하는 것으로는 다소 무리가 있음(단, 엔터프라이즈 요금제와 일반사용자 요금제 밖에 없으므로 아직까지 고려사항이 아니라고 판단됨.).

—----------------------------------------------------------------

회의 내용

1. 데이터셋 확보 문제 따른 LLM 개발 방안 모색

-> 협의 내용 : 데이터셋 확보에 어려움이 있으니, <https://annajin.tistory.com/226>  
 과 같은 프롬프팅 - 답변 기반의 형식도 고려해볼 필요 있을 것 같다.

우선, 최대한 데이터셋을 확보할 수 있도록 하되, 확보가 어려운 것으로 판단되면

프롬프팅을 이용한 질의 응답 기반 작동 하는 방식도 고려해 보기.

단, 정확도 및 안정성 측면을 고려해서 어떻게 제어할 수 있을지 의견 내보자.

1. 개발 환경 공유

-> jdk 및 IDE 개발 도구 공유 필요 있음. 우선 개발하면서 계속해서 적어주기로 했고, 추후 의견 모아서 통합하기로 했습니다.

<< 아직 어떤 버전으로 개발할 지 미정이고,

추후 테스팅 + 추가적인 회의 통해서 업데이트 할 예정입니다.

1. DB 설계

-> 개발하면서 의견 조율하기로 결정했습니다.