

Gemini API를 활용한 웹 취약점 분석 프로그램

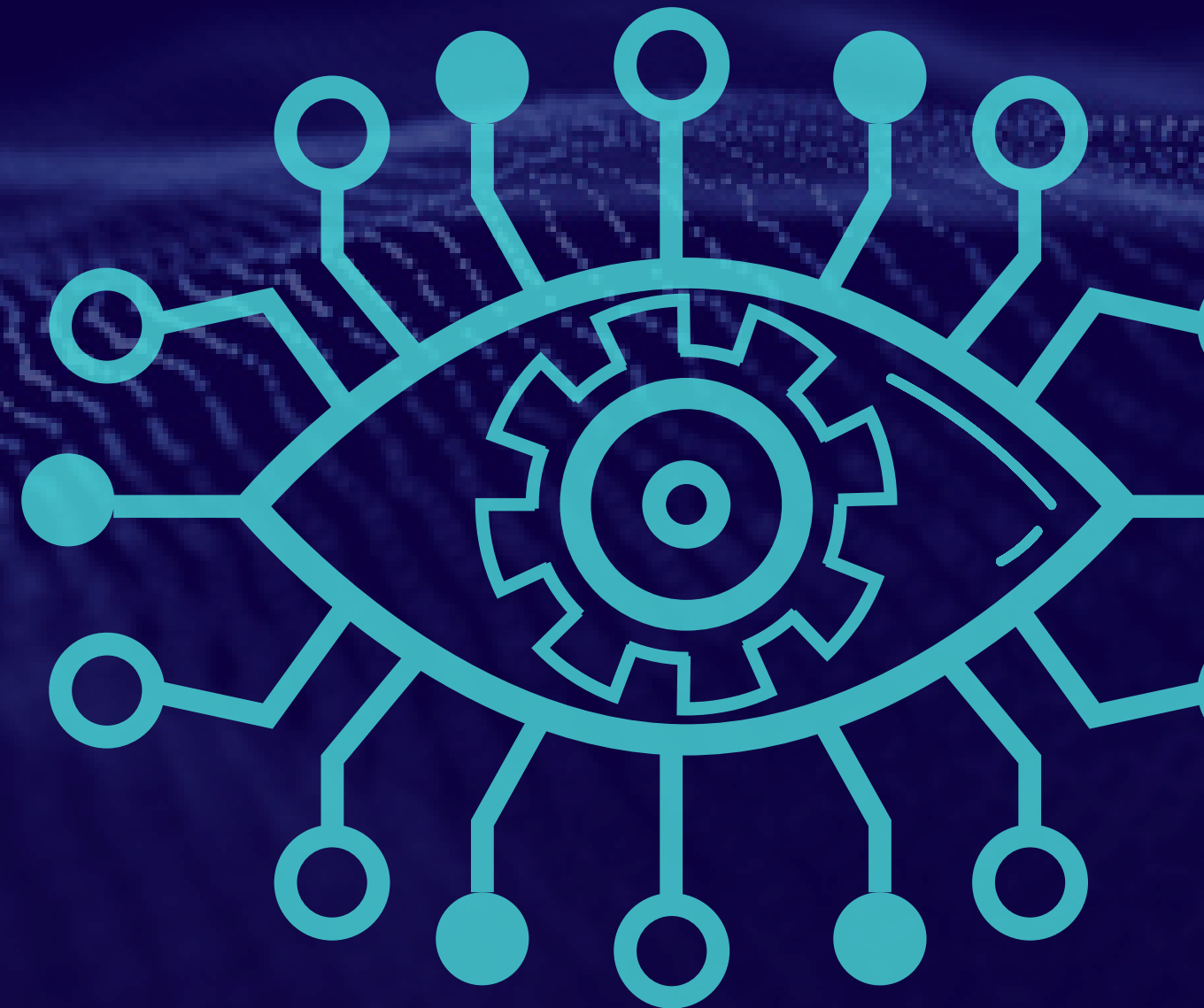
10210 이수빈 / 10211 이승훈



넬포유 7월 내부 세미나

CONTENTS

- 프로젝트 소개
- 개발 동기
- 목표
- 기술 스택
- 시스템 구조
- 주요 기능
- 구현과정
- 핵심 동작 원리
- 배포 환경
- 기대 효과
- 한계점
- 발전 방향
- 정리



Gemini API를 활용한 웹 취약점 분석 프로그램

Gemini API를 활용하여 웹사이트의 입력값과 응답 데이터를 분석하고, 웹 취약점을 탐지하는 AI기반 취약점 분석 프로그램이다.

데이터 유출 비용 증가

2023년 11월 14일, IBM 시큐리티의 '2023 데이터 유출 비용 연구 보고서'에서는 데이터 유출로 인한 전 세계 평균 비용이 지난 3년간 15% 증가한 약 445만 달러로 올해 사상 최고치를 기록했다고 밝혔다. 한국 역시 데이터 유출로 인한 평균 비용이 3년간 19% 증가하며 45억 3600만원으로 사상 최고치를 기록했다는 내용 또한 제기되었다.

이에 따른 영향

문제는 공개된 데이터 침해 건수 중 33%만이 실제 기업 내부 보안 팀에서 밝혀냈다는 점이다. 침해 사고의 40%는 제3자에 의해, 나머지 4분의 1 이상은 랜섬웨어 공격자에 의해 공개되었다.

주목할 또다른 사항은 데이터 유출은 기업을 넘어 소비자에게까지도 영향을 미친다는 점이다. 조사에 참여한 기업의 57%가 데이터 유출 비용 상승으로 인해 서비스 및 제품의 가격을 인상했다고 답했으며 발생 후 51%의 조직이 보안 지출을 늘린 것으로 나타났다.

목표

Gemini API를 활용한 웹 취약점 분석 프로그램

1

본 프로젝트는 Google Gemini API를 활용하여 웹 애플리케이션의 보안 취약점을 자동으로 분석하고 결과를 제공하는 보안 분석 도구를 개발하는 것을 목표로 한다. 입력된 웹 코드 또는 URL을 기반으로 SQL Injection, XSS, CSRF 등의 취약점을 탐지하고, AI가 그 원인과 공격 방식, 해결 방법을 설명한다. 또한 개발자가 쉽게 이해하고 적용할 수 있도록 보안 개선 예시와 함께 리포트를 제공하며, 향후 기능 확장을 고려한 구조로 설계한다.

2

이 프로젝트는 Gemini API를 활용해 웹 코드의 보안 취약점을 자동으로 식하고 쉽게 설명해주는 도구를 만드는 것을 목표로 한다. SQL Injection, XSS 같은 문제를 찾아내고 그 원인과 해결 방법을 AI가 자연어로 설명해 개발자가 이해하고 적용할 수 있도록 한다. 보안 학습과 실습을 동시에 지원하는 도구로 설계하였다.

기술 스택

프론트 엔드

HTML, CSS,
JavaScript,
Json

백엔드

Node JS

시스템 구조

- 사용자 입력 UI 제공 (웹 코드 / URL 입력)
- 분석 요청 버튼
- 결과 리포트 출력 화면
- 기술: HTML / CSS / (선택: JS)

서버

- 프론트엔드 요청 처리
- 입력 데이터 검증 및 전달
- Gemini API 요청 관리
- 결과 받아서 정리 후 반환
- 기술: Node.js / Python Flask
가능

모듈

- 입력 데이터 전처리
- 취약점 분석 프롬프트 생성
- 분석 결과 구조화 처리

- 분석 결과를 보기 쉽게 정리
- 취약점 종류별 분류
- 개선 코드 예시 포함
- 사용자에게 최종 결과 제공

주요 기능

입력된
웹 코드 또는
URL을 기반으로
분석 수행

Gemini API를
활용하여 탐지
결과를 자연어로
설명

취약한 코드에
대한 수정 방법
제시

분석 결과를
보기 쉽게
구조화

웹 URL 입력
또는 코드 직접
입력 지원

향후 다양한
언어(PHP, JS,
Python 등)
지원 가능

**웹 취약점
자동 분석**

**AI 기반
취약점 설명**

**보안 개선
방법 제공**

**리포트
자동 생성**

**입력 방식
다양화**

**확장 가능한
분석 구조**

구현 과정

입력 및 데이터 수집

사용자가 웹 코드 또는 URL을 입력하면 이를 프론트엔드에서 받아 백엔드로 전송한다.
백엔드는 입력 데이터를 정리하고 분석 가능한 형태로 변환한다.

Gemini API 기반 취약점 분석

정제된 데이터를 Gemini API로 전달하여 SQL Injection, XSS, CSRF 등의 취약점을 분석한다.
AI가 취약점의 원인, 공격 방식, 영향 범위까지 자연어로 생성한다.

결과 처리 및 리포트 출력

분석 결과를 백엔드에서 구조화하여 정리한 뒤 프론트엔드로 전달한다.
사용자는 이를 리포트 형태로 확인하고, 취약점과 개선 방법을 쉽게 이해할 수 있다.

핵심 동작 원리



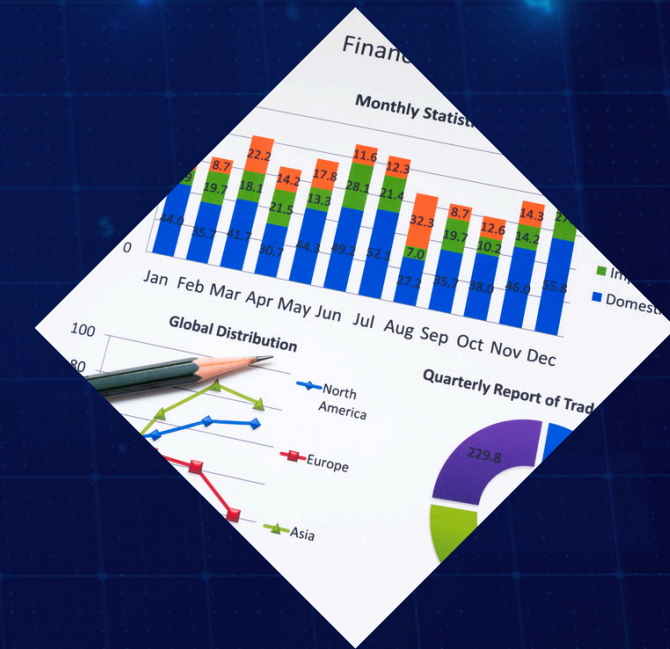
입력 데이터 기반 분석 구조

- 사용자가 입력한 웹 코드 또는 URL을 기반으로 시스템이 분석 대상을 정의하고, 보안 취약점 검사를 위한 데이터를 준비한다.



AI(Gemini) 기반 패턴 분석

- Gemini API가 입력 데이터를 분석하여 SQL Injection, XSS, CSRF 등 보안 취약점을 패턴 기반으로 탐지하고, 그 원인과 위험성을 판단한다.



자연어 리포트 생성 방식

- 분석 결과를 단순 코드가 아닌 자연어 설명으로 변환하여, 취약점의 의미와 해결 방법을 사용자가 쉽게 이해할 수 있는 형태로 제공한다.

시연 결과

웹 취약점 탐지 결과 확인

사용자가 입력한 웹 코드 또는 URL을 분석한 결과, SQL Injection, XSS 등 주요 보안 취약점이 실제로 탐지되는 것을 확인할 수 있었다. 이를 통해 시스템이 기본적인 웹 보안 문제를 정상적으로 식별함을 검증하였다.

AI 기반 설명 및 개선안 출력

탐지된 취약점에 대해 Gemini API가 원인, 공격 방식, 영향 등을 자연어로 설명하고, 안전한 코드 수정 방법까지 함께 제공하는 것을 확인하였다. 이를 통해 단순 탐지가 아닌 실질적인 보안 학습 도구로 활용 가능성을 검증하였다.

배포 환경

로컬 개발 환경

개발 단계에서는 개인 PC에서 프로그램을 실행하여 테스트한다. 프론트엔드와 백엔드를 로컬 서버로 구성하고, Gemini API를 연동하여 기능을 검증한다. 주로 기능 개발, 디버깅, API 테스트 용도로 사용된다.

웹 배포 환경

완성된 시스템은 외부에서 접근 가능하도록 웹 서버에 배포한다. 클라우드 서버 또는 호스팅 서비스를 이용하여 프론트엔드와 백엔드를 운영하고, 사용자가 URL을 통해 접속하여 분석 기능을 사용할 수 있도록 한다.

기대 효과

SQL Injection, XSS 등 주요
웹 취약점을 실제 사례 기반으로
확인할 수 있어 보안 개념 이해도가
향상된다.

AI가 제공하는 취약점 원인과
수정 방법을 통해 안전한 코드 작성
습관을 기를 수 있다.

기대 효과

수동으로 진행하던 보안 점검을
자동화하여 시간과 노력을
절약할 수 있다.

단순 분석 도구가 아닌 학습용
시스템으로 활용되어,
보안 교육 및 포트폴리오 제작에
활용 가능하다.

한계점

AI 분석 정확도의 한계

Gemini API 기반 분석은 패턴과 문맥에 의존하기 때문에, 모든 웹 취약점을 100% 정확하게 탐지하는 데에는 한계가 있다. 특히 복잡한 로직이나 커스텀 구조에서는 오탐 또는 미탐이 발생할 수 있다.

실제 실행 환경 검증 부족

이 시스템은 코드 분석 중심으로 동작하기 때문에, 실제 서버 환경에서의 공격 가능성이나 동작 결과까지 완전히 검증하지는 못한다. 따라서 실제 보안 테스트 도구와 병행 사용이 필요하다.

발전 방향

다양한 보안
취약점 탐지
확장

다양한 웹 개발
언어 분석
기능 추가

개발 편의성을
높이기 위한
안전한 코드
자동변환

최종완료
입력된 URL을
실제로 크롤링해
개선

취약점의
위험도
기준 웹사이트
보안 점수
산출

개발자 학습용
플랫폼 확장

다양한 취약점
탐지 확장

다중 프로그래밍
언어 지원

자동 수정 코드
생성 기능

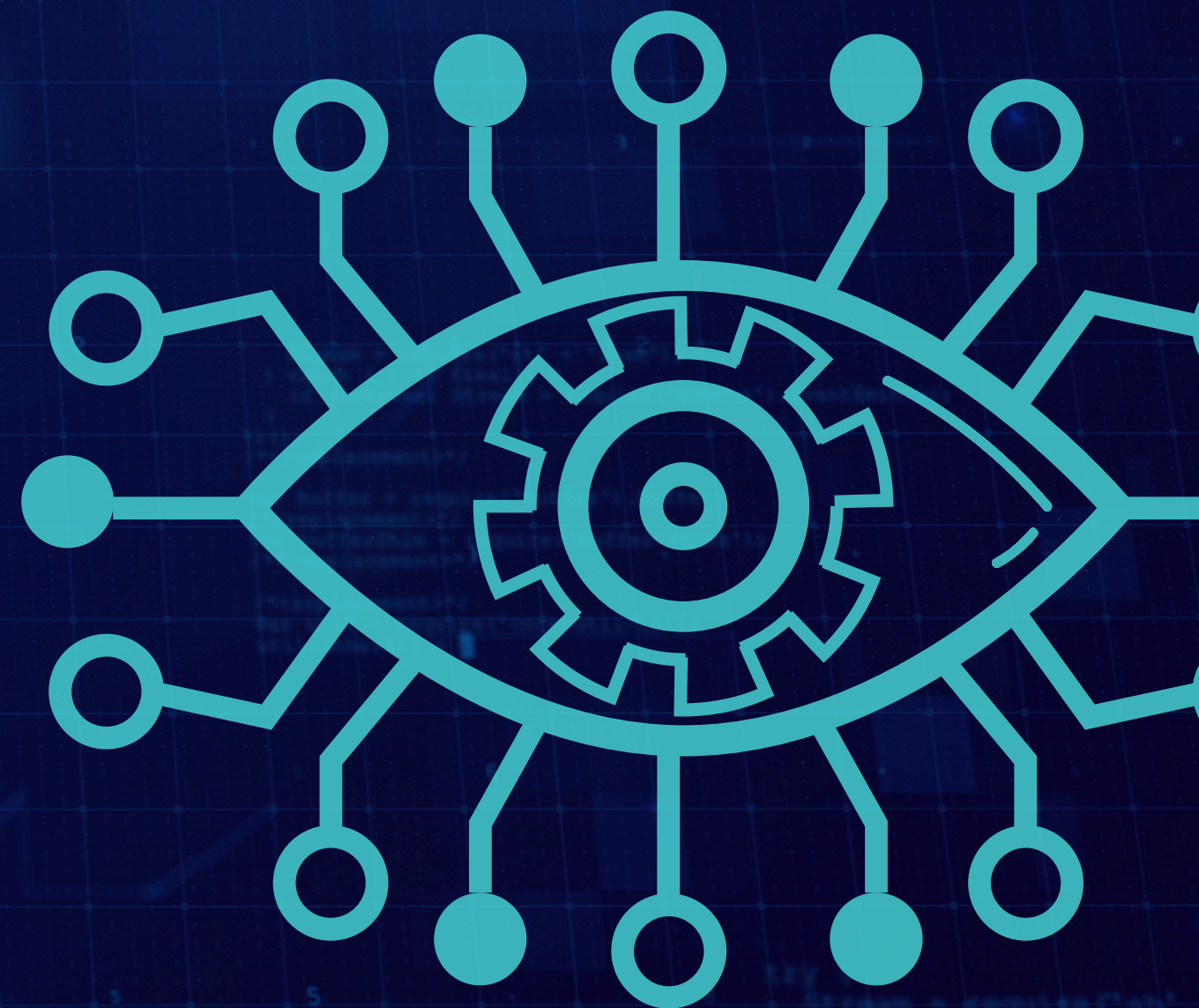
실시간 웹
크롤링 분석

보안 점수
시스템 도입

교육용 보안
플랫폼 확장

Q&A

궁금한 부분을 질문해 주세요



감사합니다

세명컴퓨터 고등학교

스마트 보안 솔루션과 1학년 2반

