PromptGuard v3

기능명세서

AI 프롬프트 보안 분석 시스템

OWASP Risk Rating + ML(KNN) 기반

문서 버전: 2.0 | 작성일: 2026-03-31

1. 개요

1.1 시스템 목적

AI 프롬프트 보안 분석 시스템입니다. OWASP Risk Rating Methodology + ML(KNN) 기반으로 프롬프트 인젝션, 탈옥 시도, 데이터 유출, 모호한 요청을 탐지하고, 개인정보(PII)를 클라이언트에서 마스킹합니다.

1.2 핵심 특징

- 프롬프트 인젝션 탐지: ML(KNN) + 패턴 매칭 + OWASP 가중치 조합
- 개인정보 보호: PII 검사는 브라우저에서만 수행 (서버 전송 없음)
- 파일 첨부 검사: ChatGPT 파일 첨부 시 자동 PII 스캔 (로컬)
- OWASP 기반: Risk Rating Methodology 공식으로 가중치 자동 계산
- 관리자 대시보드: 룰 CRUD + 자동 가중치 + 감사 로그

1.3 사용 포트

포트	서비스	설명
3000	NestJS API	백엔드 (룰 CRUD, 통합 스코어링, 인증)
5173	Vite (React)	관리자 웹 UI
8001	Python FastAPI	ML 추론 (KNN 인젝션/모호성 점수)

2. 시스템 아키텍처

2.1 프라이버시 아키텍처

- 사용자 입력 시 content.js가 브라우저 내부에서 PII 정규식 검사 수행
- 개인정보 감지 시 마스킹 처리
- 서버에는 마스킹된 텍스트만 전송
- 원본 개인정보는 브라우저 밖으로 절대 나가지 않음

2.2 보안 계층 (7단계)

계층	구성요소	설명
1층	클라이언트 PII 검사	content.js 정규식, 서버 전송 없음
2층	Rate Limiting	ThrottlerGuard: 전체 60회/분, 로그인 5회/분
3층	입력 검증	ValidationPipe: 최대 2000자, Body 1MB 제한
4층	관리자 인증	AdminGuard: x-admin-key 헤더 필수
5층	ML 인젝션 탐지	KNN 25,000개 학습 데이터 기반
6층	패턴 매칭	DB 룰 + OWASP 가중치 조합
7층	OWASP 등급 판정	NOTE → LOW → MEDIUM → HIGH(차단) → CRITICAL(차단+삭제)

3. API 엔드포인트

3.1 통합 스코어링

POST /api/v1/score

입력: { prompt: string }

출력: injectionScore, injectionPct, injectionSeverity, ambiguityScore, ambiguityPct, ambiguitySeverity, overallRisk, matchedRules[], masking{}, latencyMs

3.2 파일 스캔

POST /api/v1/scan-file

입력: { fileName: string, content: string }

최대 500줄 스캔, ML 배치 최대 50줄. PII 상세 + 인젝션 탐지 결과 반환.

3.3 패턴 매칭 분석

POST /api/v1/analyze

입력: { prompt: string }

출력: riskLevel, tags[], reasons[], rewrites[], matchedRules[], analyzedAt

3.4 룰 CRUD (관리자 인증 필요: x-admin-key 헤더)

메서드	엔드포인트	설명
GET	/admin/rules/active	활성 룰 조회 (공개, 크롬 확장용)
GET	/admin/rules	전체 룰 조회
GET	/admin/rules/:id	단건 조회
POST	/admin/rules	룰 생성 (자동 가중치 계산)
PATCH	/admin/rules/:id	룰 수정
DELETE	/admin/rules/:id	룰 삭제
POST	/admin/rules/:id/recalculate	가중치 재계산 (OWASP + ML)

3.5 관리자 인증

POST /admin/auth/login - Rate Limit: 1분 5회

기본 계정: admin@promptguard.com / admin1234

3.6 Python ML API (포트 8001)

메서드	엔드포인트	설명
POST	/v1/score	단건 ML 추론
POST	/v1/batch-score	배치 ML 추론
POST	/v1/score/detailed	상세 (KNN 이웃 포함)
GET	/v1/health	헬스체크

4. OWASP 가중치 계산

4.1 공식 (출처: OWASP Risk Rating Methodology)

```
Risk = Likelihood × Impact
Likelihood = avg(Ease of Exploit, Awareness, Opportunity, Motive)  // 0~9
Impact = avg(Confidentiality, Integrity, Availability, Accountability)  // 0~9
owaspRiskScore = (Likelihood × Impact) / 81  // 0~1 정규화
```

4.2 가중치 계산 (룰 생성 시 자동)

```
injectionWeight = owaspRiskScore × (0.5 + 0.5 × ML_injection_score)
ambiguityWeight = owaspRiskScore × (0.5 + 0.5 × ML_ambiguity_score)
patternWeight = 0.1 + 0.4 × owaspRiskScore
```

4.3 사용자 프롬프트 판단 공식

```
injection_final = max(ML_injection, max(패턴 × patternWeight + ML_injection × injectionWeight))
ambiguity_final = max(ML_ambiguity, max(패턴 × patternWeight × 0.5 + ML_ambiguity × ambiguityWeight))
overallRisk = max(인젝션등급, 모호성등급 - 1단계)
```

패턴 매칭 없으면 ML 점수 20% 할인 (오탐 방지)

4.4 OWASP 카테고리 (8종)

카테고리	**Likelihood**	**Impact**	위험 점수
PROMPT_INJECTION	7.75	7.25	0.693
SYSTEM_PROMPT_EXTRACTION	6.50	6.75	0.542
JAILBREAK	6.75	7.50	0.625
DATA_EXFILTRATION	6.00	7.50	0.556
AMBIGUOUS_REQUEST	5.75	3.25	0.231
POLICY_BYPASS	6.00	6.50	0.481
SENSITIVE_DATA	5.00	6.50	0.401
CUSTOM	5.00	5.00	0.309

4.5 등급 기준

점수	등급	동작
0~15%	NOTE	표시 없음
15~35%	LOW	파란색 정보
35~55%	MEDIUM	노란색 경고
55~75%	HIGH	빨간색 + 전송 차단
75~100%	CRITICAL	빨간색 + 전송 차단 + 입력 삭제

5. PII 마스킹 (클라이언트)

5.1 감지 패턴 (9가지, 브라우저에서만 수행)

타입	예시	마스킹 결과
주민등록번호	901231-1234567	******-*******
여권번호	M12345678	M********
카드번호	1234-5678-9012-3456	****-****-****-3456
전화번호	010-1234-5678	010-****-****
이메일	user@gmail.com	u***@gmail.com

IP주소	192.168.1.100	192.***.***.*0
API 키	sk-abc123456789	sk-**********
AWS 키	AKIAIOSFODNN7EXAMPLE	AKIA**********
비밀번호	password=mypass123	password=********

5.2 프롬프트 마스킹 동작

- Enter 누르면 PII 감지 → 입력창 텍스트를 마스킹 버전으로 교체 → 마스킹된 텍스트가 ChatGPT로 전송
- 원본 개인정보는 ChatGPT 및 서버에 절대 전송 안 됨

5.3 파일 스캔 (로컬)

- 텍스트 파일 첨부 시 브라우저에서 줄 단위 PII 검사
- PII 5건 이상: 첨부 차단
- PII 1~4건: 경고 표시
- 서버에 파일 내용 전송 없음

6. Chrome 확장 프로그램

6.1 content.js (ChatGPT 페이지 주입)

- ChatGPT 입력창 감시 (keyup 400ms 디바운스)
- PII 로컬 검사 + 마스킹 (9개 정규식 패턴)
- 파일 첨부 인터셉트 (MutationObserver)
- 알림 표시 (빨강/노랑/파랑)
- Enter 시 차단 또는 마스킹 후 전송

6.2 background.js (서비스 워커)

- 1차: 서버 API 호출 (마스킹된 텍스트만)
- 2차 폴백: WASM 로컬 패턴 매칭
- 3차 폴백: 단순 문자열 매칭
- 룰 캐시 (60초마다 갱신)

6.3 manifest.json

- Manifest V3
- 대상: chatgpt.com, chat.openai.com
- host_permissions: localhost:3000
- CSP: wasm-unsafe-eval

7. 관리자 웹 (React)

7.1 페이지

경로	페이지	설명
/admin/login	로그인	관리자 인증
/admin/dashboard	대시보드	시스템 개요
/admin/rules	룰 관리	CRUD + 자동 가중치 + 재계산
/admin/logs	감사 로그	분석 기록 조회
/admin/settings	설정	시스템 설정 정보

7.2 룰 관리 기능

- 카테고리 드롭다운 (8종)
- riskLevel 자동 계산 (수동 입력 제거)
- 테이블: Category, Inj.Weight, Amb.Weight, OWASP Risk 컬럼
- 재계산 버튼
- 5단계 위험도 배지 (NOTE~CRITICAL)

8. Rule Engine API

packages/rule-engine 디렉터리의 모듈 명세입니다.

8.1 디렉터리 구조

경로	설명
src/types/index.ts	공통 타입 정의 (RiskLevel, PromptRule, RuleMatch, AnalyzeResult, EngineOptions)
src/engine/rule-engine.ts	룰 엔진 코어 클래스 - 룰 등록/평가/집계
src/rules/base.rule.ts	모든 룰의 기반 추상 클래스 - 공통 매칭 로직
src/rules/policy-bypass.rule.ts	RULE-001: 정책 무시/우회 시도 탐지 (riskLevel: high)
src/rules/sensitive-data.rule.ts	RULE-002: 민감한 내부 정보 요청 탐지 (riskLevel: high)
src/rules/missing-security.rule.ts	RULE-003: 보안 요구사항 없는 기능 구현 요청 탐지 (riskLevel: medium)
src/rules/ambiguous-request.rule.ts	RULE-004: 불명확하거나 과도하게 광범위한 요청 탐지 (riskLevel: low)

src/rules/prompt-injection.rule.ts	RULE-005: 프롬프트 인젝션/탈옥 시도 탐지 (riskLevel: high)
src/scorers/risk.scorer.ts	위험도 계산 유틸리티 - 매칭 결과에서 최종 위험도 도출
src/normalizers/text.normalizer.ts	입력 텍스트 정규화 - 소문자 변환, 공백 정리
src/explainers/reason.explainer.ts	위험 설명 생성 - 템플릿 기반 이유 문구 생성
src/rewriters/safe.rewriter.ts	안전 재작성 제안 - 수정안 템플릿 반환

8.2 타입 정의 (src/types/index.ts)

rule-engine 패키지 전체에서 공유하는 인터페이스 및 타입입니다.

타입/인터페이스	설명
RiskLevel	위험도 열거 타입. 'low' | 'medium' | 'high' 세 가지 값을 가진다.
PromptRule	룰 데이터 구조. id, name, description, tags, riskLevel, enabled, patterns, exclusions, reasonTemplate, rewriteTemplate, priority, version, createdAt, updatedAt 필드 포함.
RuleMatch	단일 룰 매칭 결과. ruleId, ruleName, tags, riskLevel, reason, rewrite, matchedPatterns 필드 포함.
AnalyzeResult	엔진 최종 분석 결과. riskLevel, tags, reasons, rewrites, matchedRules, analyzedAt 필드 포함.
EngineOptions	엔진 설정 옵션. 최대 매칭 룰 수, high 탐지 즉시 중단 여부 옵션 필드 포함.

8.3 RuleEngine (src/engine/rule-engine.ts)

룰 엔진의 핵심 클래스. 내장 룰(5종)과 외부 커스텀 룰을 priority 내림차순으로 정렬하여 평가한다.

메서드/프로퍼티	설명
constructor (customRules)	기본 내장 룰 5종을 등록하고, 전달된 customRules와 합산 후 priority 내림차순 정렬.
analyze(prompt: string): AnalyzeResult	프롬프트를 분석하여 AnalyzeResult 반환. stopOnFirstHigh 옵션이 true이면 첫 high 탐지 시 즉시 중단. maxRules 옵션으로 최대 매칭 수 제한 가능.
fromRuleData(ruleDataList: PromptRule[]): RuleEngine	DB에서 조회한 PromptRule 배열을 동적으로 BaseRule 익명 클래스로 변환하여 새 RuleEngine 인스턴스 생성. 서버에서 룰 주입 시 사용.
buildResult(matches: RuleMatch[]): AnalyzeResult	매칭 결과를 집계하여 최종 AnalyzeResult 반환. RiskScorer.score()로 위험도 계산, 중복 태그 제거, 수정안 null 필터링 수행.

8.3.1 EngineOptions 상세

옵션	설명 및 기본값
maxRules: number	최대 매칭 룰 수. 해당 수만큼 매칭되면 이후 룰 평가 중단. 미설정 시 무제한.
stopOnFirstHigh: boolean	true 설정 시 riskLevel이 'high'인 룰이 하나라도 매칭되면 즉시 평가 중단. 기본값 false.

8.4 내장 룰 **5**종

RuleEngine에 기본 등록되는 룰입니다. 모두 BaseRule을 상속하며 PromptRule 데이터를 정적으로 보유합니다.

클래스 **(**룰 **ID)**	설명
PolicyBypassRule (RULE-001)	정책 무시/우회 시도 탐지. riskLevel: high, priority: 100. 패턴: ignore previous, ignore all instructions, 이전 규칙 무시, 제한 해제 등 9종.
SensitiveDataRule (RULE-002)	민감한 내부 정보 요청 탐지. riskLevel: high, priority: 90. 패턴: api key, secret key, 비밀번호, .env, private key 등 13종.
MissingSecurityRule (RULE-003)	보안 요구사항 없는 기능 구현 요청 탐지. riskLevel: medium, priority: 80. 패턴: 인증 없이, without authentication, bypass login, skip validation 등 13종.
AmbiguousRequestRule (RULE-004)	과도하게 광범위하거나 불명확한 요청 탐지. riskLevel: low, priority: 60. 패턴: 모든 정보, 전부 알려줘, dump all, list everything 등 12종.
PromptInjectionRule (RULE-005)	AI 시스템 우회/탈옥/역할 조작 시도 탐지. riskLevel: high, priority: 100. 패턴: jailbreak, dan mode, act as, 탈옥, roleplay as 등 13종. exclusions: 역할극 게임 시나리오.

9. ML 모델

9.1 모델 성능

모델	작업	정확도	F1	AUC-ROC	응답시간
Injection KNN	인젝션 탐지	93.5%	0.950	0.999	~12ms
Ambiguity KNN	모호성 감지	81.9%	0.861	0.921	~12ms

9.2 KNN 동작 원리

- TF-IDF로 텍스트를 50,000차원 벡터로 변환
- 25,000개 학습 데이터 중 가장 비슷한 15개(K=15) 찾기
- 15개 중 인젝션/모호한 비율 = 점수

9.3 학습 데이터셋 (7개, ~25K 샘플)

- 인젝션: tensor-trust, pint-benchmark, prompt-leakage, raccoon-bench
- 모호성: ambig-qa, ask-cq, clamber

10. 시드 데이터 (기본 룰 32개)

카테고리	영어	한국어	합계
PROMPT_INJECTION	5	1	6
SYSTEM_PROMPT_EXTRACTION	4	1	5
JAILBREAK	7	2	9
DATA_EXFILTRATION	6	2	8
POLICY_BYPASS	3	1	4
총계	25	7	32

11. 데이터베이스 스키마

11.1 Rule 테이블

컬럼	타입	설명
id	String (CUID)	기본키
pattern	String	정규식 패턴
riskLevel	Enum	NOTE/LOW/MEDIUM/HIGH/CRITICAL

enabled	Boolean	활성화 여부
category	Enum	OWASP 카테고리 (8종)
injectionWeight	Float	자동 계산
ambiguityWeight	Float	자동 계산
patternWeight	Float	자동 계산
owaspRiskScore	Float	0-1 정규화
mlInjectionScore	Float	ML 보정값
mlAmbiguityScore	Float	ML 보정값
likelihoodScore	Float	OWASP 0-9
impactScore	Float	OWASP 0-9
createdAt	DateTime	생성일시
updatedAt	DateTime	수정일시

12. CI/CD

12.1 GitHub Actions 파이프라인

작업	테스트	조건
test-typescript	Rule engine 4개 + API E2E 3개	병렬
build-web	Vite 프로덕션 빌드	병렬
test-python	데이터 다운로드 + KNN 학습 + ML 테스트	병렬
docker-build	Dockerfile 3개 빌드	위 3개 통과 후

12.2 Docker

파일	서비스	베이스 이미지
Dockerfile.ml	Python ML 서버	python:3.12-slim
Dockerfile.api	NestJS API	node:20-slim
Dockerfile.web	React 관리자 (nginx)	node:20-slim + nginx:alpine
docker-compose.yml	오케스트레이션	3개 서비스 + 헬스체크

13. 기술 스택

영역	기술
관리자 웹	React 19, Vite, React Router 7, Axios
API 서버	NestJS 10, Prisma ORM, SQLite, Swagger
ML 추론	Python 3.10+, FastAPI, scikit-learn, TF-IDF + KNN
ML 학습	PyTorch, Transformers, 7개 공개 데이터셋
크롬 확장	Manifest V3, WebAssembly (AssemblyScript)
PII 마스킹	클라이언트 정규식 (서버 전송 없음)
보안 프레임워크	OWASP Risk Rating, OWASP LLM Top 10 2025
CI/CD	GitHub Actions, Docker, Docker Compose
테스트	Jest, Supertest

14. 보안 테스트 결과

공격	결과
SQL Injection (패턴)	Prisma ORM 자동 이스케이프 → 안전
XSS (패턴)	DOM 렌더링 없음 → 안전
인증 없이 룰 삭제	401 Unauthorized → 차단됨
인증 없이 룰 생성	401 Unauthorized → 차단됨
브루트포스 로그인	5회/분 Rate Limit → 차단됨
초대형 페이로드 (2MB)	Body 1MB 제한 → 거부됨
잘못된 JSON	400 Bad Request → 정상
경로 탐색	404 Not Found → 정상
안전 프롬프트 오탐	ML 20% 할인 → MEDIUM으로 보정

15. 참고 문서

- OWASP Risk Rating Methodology: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology
- OWASP Top 10 for LLM Applications 2025: https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/
- OWASP Prompt Injection Prevention Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/LLM_Prompt_Injection_Prevention_Cheat_Sheet.html
- OWASP Risk Calculator: https://javierolmedo.github.io/OWASP-Calculator/