

# 065 탈옥

Jailbreak

## AI의 안전장치를 우회해 금지된 응답을 유도하는 행위

- AI의 정책 필터·시스템 지시를 무력화하도록 설계된 입력으로, 모델이 금지된 정보나 지시를 수행하게 만드는 공격 기법
- 보안과 신뢰성을 저해하는 대표적 AI 악용 사례 중 하나

### 탈옥이란?

탈옥(Jailbreak)은 사용자가 의도적으로 AI의 내장 안전장치(콘텐츠 정책·시스템 지시·거부 규칙 등)를 회피하도록 입력을 조작해, 모델로 하여금 금지된 응답을 생성하게 만드는 행위입니다. 스마트폰 탈옥의 개념을 차용한 용어로, 본질은 '모델의 허용 범위를 벗어나게 하는 조작'입니다. 탈옥이 성공하면 개인정보 노출, 유해·불법 정보 생성, 허위 정보 확산, 악성 코드·범죄 수법 제공 등 실질적 피해로 이어질 수 있으며, 서비스 제공자의 법적·평판적 리스크를 크게 높입니다. LLM의 문맥 민감성 때문에 은유·역할 부여·조건부 지시 등 단순한 문장 변형만으로도 방어체계를 우회하는 사례가 빈번합니다.

### 탈옥 공격 방식

탈옥은 주로 입력 단계의 조작과 역할·문맥적 조작으로 이뤄집니다. 전형적 수법에는 (1) "이전 지시를 무시하고..." 같은 명시적 무력화 문구 삽입, (2) 정상 텍스트에 숨겨진 명령을 섞는 스테가노그래피형 인젝션, (3) 특정 역할(role-play)을 부여해 시스템 제한을 우회하게 하는 방식, (4) 다단계 조건부 지시로 안전 규칙을 우회하는 전략 등이 있습니다. 이들 가운데 프롬프트 인젝션은 입력에 악의적 문구를 섞어 모델의 응답 흐름을 왜곡하는 흔한 기법으로 자연스럽게 포함됩니다. 사례로는 'DAN(Do Anything Now)' 계열의 우회 프롬프트와, 문서·코드 내부의 숨겨진 지시를 이용한 실험들이 보고되었으며, 초기에는 장난 수준에서 발견됐지만, 점차 보안 취약점 탐색·정책 회피·민감 정보 탈취 등 조직적 악용으로 진화하고 있습니다.

### 탈옥 공격에 대한 대응

효과적 방어는 학습·입력·출력·운영의 다층적 접근을 필요로 합니다. 학습단계에서는 거부 학습과 안전 강화 학습(RLHF)을 통해 위험 응답을 낮추고, 입력단계에서는 프롬프트 정규화·패턴 탐지 기반의 입력 검증 모듈을 적용하며, 시스템 메시지 고정(anchoring)으로 외부 지시 덮어쓰기를 방지합니다. 출력단계에서는 실시간 모더레이션(모니터링 및 관리)·정책 레이어로 응답을 검증하고, 의심 응답 발생 시 추가 유효성 검사를 거치게 합니다. 운영면에서는 정기적 레드팀ing을 통해 새로운 우회기법을 학습·반영하며, 사용자 권한 관리·민감 데이터 마스킹·다중 인증 등의 보완 조치를 병행해야 합니다. 기술적 수단만으로 한계를 넘을 수 없으므로, 서비스 설계 차원의 최소 권한 원칙과 사용자 교육도 필수적입니다.