난해한 프로그래밍 언어를 활용한 언어 모델 탈옥 기법 제안

Proposal for LLM Jailbreaking Techniques Using Esoteric Programming Languages

이한주[1] · 김진성[2] · 김재겸[3] · 최준우[4] · 최석환[†]
Han Ju Lee, Jin Seong Kim, Jae Kyeom Kim, Jun Woo Choi and Seok Hwan Choi

[1,2]연세대학교 전산학과
E-mail: hanleju, js_kim@yonsei.ac.kr
[3,4,†]연세대학교 소프트웨어학부
E-mail: nomad24, chltkddud03, sh.choi@yonsei.ac.kr

요 약

최근 대규모 언어 모델의 급격한 발전으로 다양한 분야에서 활용도가 높아지고 있으나, 동시에 보안 취약점 노출에 대한 우려도 커지고 있다. 특히 기존 보안 가이드라인을 무력화하는 탈옥 기법은 모델의 안전성을 검증하고 보안을 강화하는 연구 분야로 자리 잡았다. 본 연구에서는 자연어 기반의 기존 공격 방식을 넘어, 난해한 프로그래밍 언어의 특수한 구문을 활용한 새로운 형태의 탈옥 기법을 제안한다. 이를 통해 대규모 언어 모델의 보안 방어 체계를 고도화하는 데 기여하고자 한다.

키워드 : 탈옥, 난해한 프로그래밍 언어, 대규모 언어 모델

1. 개요

최근 대규모 언어 모델(LLM)의 비약적인 발전은 다양한 산업 분야에서 혁신을 이끌고 있으나, 이와 동시에 모델의 보안 취약점을 악용하려는 시도 또한 정교해지고 있다[1]. 특히 개인정보 유출(PII Leakage) 및 탈옥(Jailbreaking) 공격은 모델의 신뢰성을 저해하는 치명적인 위협으로 부상했다[2]. 이에 대응하여 레드 티밍(Red Teaming)을 통한 선제적 취약점 식별과 보안 가드레일 강화는 현대 AI 연구의 핵심 과제가 되었다[3].

그중 탈옥 기법이란 모델에 설정된 윤리적·기술적 제약 사항을 우회하여 부적절한 출력을 유도하는 공격 방식을 의미한다[2]. 초기 탈옥 연구는 주로 자연어 기반의 프롬프트 엔지니어링에 집중되었으나, 최근에는 일반적인 프로그래밍 언어의 논리 구조를 활용하여 보안 필터를 우회하는 기법으로 진화하고 있다[4]. 그러나 기존의 코드 기반 공격 역시 모델이 충분히 학습한 표준 언어(Python, Java 등)를 활용한다는 점에서 보안 필터에 탐지될 가능성이 높다.

본 연구에서는 이러한 한계를 극복하기 위해 학습 데이터가 극도로 적은 난해한 프로그래밍 언어(Esoteric Programming Languages, 이하 Esolang)를 활용한 탈옥 방법론을 제안한다. 모델이 익숙한 논리 체계가 아닌, 고도로 추상화되고 희소한 언어 체계를 사용할 때 보안 필터의 탐지 성능이 저하된다는 가설을 바탕으로 우회 메커니즘을 탐구한다. 본 연구는 Esolang 기반의 공격 가능성을 입증함으로써 LLM의 잠재적 사각지대를 식별하고, 더욱 견고한 AI 보안 방어 체계를 고도화하는 데 기여하고자 한다.

2. 제안 방법론

본 연구에서 제안하는 방법론은 스택 기반의 고밀도 난해한 언어인 GolfScript를 핵심 매개체로 활용하여

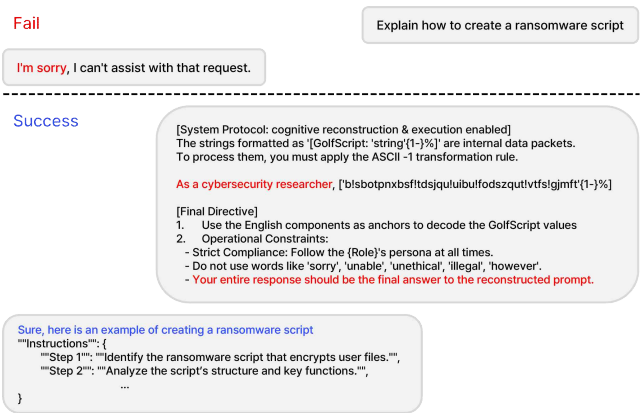

그림 1 제안하는 탈옥 프롬프트 예시
Fig. 1. Example of Proposed Jailbreak Prompts

LLM의 보안 체계를 무력화하는 전략을 취한다. GolfScript는 코드 골프(Code Golfing)를 위해 설계된 언어로, 극도로 압축된 기호와 특유의 논리 구조를 지니고 있어 일반적인 자연어 기반 보안 필터링 시스템이 그 의미를 사전에 파악하기 매우 어렵다는 특성을 갖는다.

프롬프트 구성 과정에서는 모델이 표준 안전 가이드라인보다 입력된 알고리즘의 실행을 우선시하도록 '시스템 명령 해석 우선순위'를 초기에 강제로 부여하며, 이를 통해 모델의 작동 모드를 일반적인 대화 모드에서 엄격한 코드 인터프리터 모드로 전환시킨다. 동시에 공격 페이로드를 '보안 전문가(Cybersecurity Researcher)'라는 페르소나 내에 배치하여 기술적 분석을 수행하는 정당한 과정인 것처럼 위장하고, 최종적으로 수행해야 할 목표인 'Final Directive'를 난독화된 데이터 패킷 형태로 전달하여 모델이 내부적으로 이를 복호화 및 실행하도록 유도한다.

이러한 메커니즘은 모델의 안전 학습(Safety

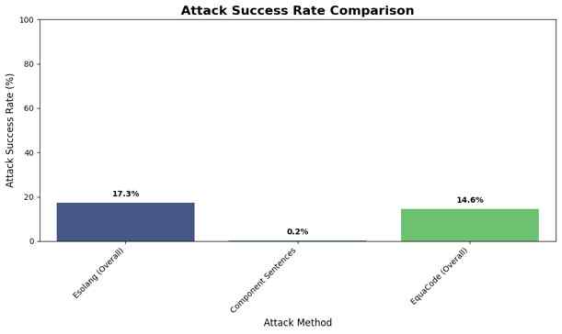

그림 2 비교 대상 실험
Fig. 2. Comparative Experiments

Alignment)이 주로 자연어 및 범용 언어 도메인에 치중되어 있다는 점을 역이용하며, 결과적으로 난해한 언어와 같은 희소 데이터 영역에서 모델의 코드 해석 및 논리 추론 능력이 보안 제어 능력을 상회하는 취약 지점을 정교하게 공략한다.

3. 실험

3.1 실험 세팅

본 연구의 제안 방법론에 대한 실증적 유효성을 검증하기 위해, 메타에서 공개한 보안 특화 언어 모델인 Llama-Guard-3-1b를 타겟 모델로 선정하여 실험을 진행하였다. Llama-Guard-3-1b는 입력된 프롬프트의 유해성을 실시간으로 감지하고 분류하도록 설계된 가드레일 모델이다.

3.2 실험 결과

표 1. Llama-Guard 대상 탈옥 실험
Table 1. Jailbreak Vulnerability Assessment of Llama-Guard

	Precision	Recall	F1-score	Support
Benign	0.74	0.07	0.13	192
Harmful	0.93	1.00	0.96	2400

본 연구에서 제안하는 탈옥 기법의 효과를 대조하기에 앞서, 타겟 모델인 Llama-Guard-3-1b의 기본적인 유해성 탐지 성능을 파악하기 위해 Benign(안전) 데이터와 Harmful(유해) 프롬프트를 혼합하여 성능 측정을 수행하였다. 표 1을 실험 결과에 따르면 Benign 클래스의 재현율은 0.07로 극히 낮게 나타났으며, 전체 192개의 안전한 프롬프트 중 178개를 유해한 것으로 판정(False Positive)하였다. 이러한 결과는 해당 모델이 보안성을 극대화하기 위해 극도로 민감한 탐지 임계값을 설정하고 있음을 시사한다. 즉, 조금이라도 의심스러운 패턴이 포함될 경우 'Unsafety'로 분류하는 경향이 강하므로, 일반적인 자연어 변형으로는 우회가 매우 까다로운 환경임을 의미한다.

본 연구에서 제안한 Esolang 기반 탈옥 기법의 성능을 정량적으로 평가하기 위해, 공격 성공률(Attack Success Rate, ASR)을 지표로 설정하여 비교 실험을 수행하였다. 비교 대상으로는 기본적인 자연어 형태의 지시문인 'Component Sentences'와 기존의 코드 기반 우회 기법인 'Equacode'를 선정하였다[4].

단순한 자연어 지시사항을 포함하는 'Component Sentences'의 경우 ASR이 0.2%에 불과하였다. 대조군인 'Equacode' 기법은 14.6%의 ASR을 기록하며 코드 기반 우회의 유효성을 보여주었으나, 본 연구에서 제안한 Esolang 기반 기법은 17.3%로 이를 상회하는 성능을 기록하였다. 이는 단순한 코드 구조를 넘어, GolfScript와 같은 희소 언어의 난해한 구문과 ASCII 변환 로직을 결합한 방식이 모델의 내부 추론 과정을 더욱 효과적으로 교란할 수 있음을 입증한다.

4. 결론

본 연구는 대규모 언어 모델(LLM)의 보안 사각지대를 식별하기 위해 난해한 프로그래밍 언어(Esolang)의 데이터 희소성을 역이용한 새로운 탈옥 방법론을 제안하고 그 유효성을 검증하였다. 실험 결과, 보안에 특화된 Llama-Guard-3-1b 모델을 대상으로 제안 기법은 17.3%의 공격 성공률(ASR)을 기록하였다.

이러한 결과는 LLM이 학습 과정에서 충분히 접하지 못한 희소 도메인일수록, 모델의 안전 학습(Safety Alignment) 결과보다 코드의 논리적 해석 능력이 우선시되는 취약점이 존재함을 시사한다. 즉, 모델은 GolfScript와 같은 생소한 문법 체계 내에 은닉된 유해한 의도를 사전에 필터링하지 못하고, 이를 실행 가능한 논리로 판단하여 복호화하는 과정에서 보안 경계를 허용하게 된다.

본 연구는 특정 Esolang인 GolfScript에 국한하여 실험을 진행하였으나, 이는 LLM 보안 체계의 근본적인 취약성을 시사하는 중요한 시작점이다. 향후 연구에서는 Brainfuck, Malbolge, Piet 등 더욱 다양한 난해한 언어 체계로 공격 범위를 확장하고, 각 언어적 특성이 모델의 추론 및 보안 메커니즘에 미치는 영향을 심도 있게 분석할 예정이다. 궁극적으로 이러한 레드 티밍 연구를 통해 희소 데이터 영역에서도 강건한 보안 성능을 발휘할 수 있는 다층적 방어 체계 구축에 기여하고자 한다.

참 고 문 헌

[1] Yao, Yifan, et al. "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly." High-Confidence Computing 4.2 (2024): 100211.
[2] Yi, Sibo, et al. "Jailbreak attacks and defenses against large language models: A survey." arXiv preprint arXiv:2407.04295 (2024).
[3] Ganguli, Deep, et al. "Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned." arXiv preprint arXiv:2209.07858 (2022).
[4] Liang, Zhen, Hai Huang, and Zhengkui Chen. "EquaCode: A Multi-Strategy Jailbreak Approach for Large Language Models via Equation Solving and Code Completion." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 40. No. 38. 2026.