LLM 환경에서의 프롬프트 인젝션 공격 동향 및 연구 대응 방안

권순용*, 김동호*, 박성식*, 이용준**
*극동대학교 해킹보안학과
**극동대학교 해킹보안학과
e-mail: dongho0112@naver.com

Prompt Injection Attack Trends and Research in LLM Environments and How to Respond

Sun-Yong Kwon*, Dong-Ho Kim*, Seong-Sik Park*, Yong-Jun Lee**
*Dept. of Hacking & Security, Far East University
**Dept. of Hacking & Security, Far East University

요 약

본 논문에서는 생성형 인공지능의 비약적인 발전으로 LLM의 사용이 광범위해짐에 따라 발생할 수 있는 여러 보안 이슈 중 프롬프트 인젝션 공격을 분석하고자 한다. 프롬프트 인젝션 공격의 선행 연구 및 공격 사례 분석을 통해 WAF(Web Application Firewall)와 역공학적 학습을 통한 프롬프트 인젝션의 대응 방안을 제안하고 있다. 하지만 프롬프트 인젝션 방어 기법에 대한 한계점이 명확하게 존재하기에 추가적인 보완책과 지속적인 연구가 필요하다.

1. 서론

생성형 인공지능(Generative AI)의 비약적인 발전은 사회 전반에 걸쳐 큰 반향을 일으키고 있다. 2022년 ChatGPT를 시작으로 Copilot, Gemini, Claude, Deepseek 등의 다양한 생성형 인공지능 서비스들이 등장하였다. 현재 인공지능 서비스는 일상생활은 물론, 산업, 교육, 의료, 공공 분야에 이르기까지 광범위한 영역에서 사용되고 있다. 이제는 단순한 보조 도구를 넘어 실질적인 의사결정이나 자동화, 콘텐츠 생산의 핵심 수단으로써 활용되고 있으며 그 가능성과 영향력 또한 나날이 커지고 있다. 또한 문법에 의해 복잡하고 정교한 자연어를 처리할 수 있는 대규모 언어 모델(Large Language Models, LLM)이 개발됨으로써 생성형 인공지능은 인간처럼 자연스러운 응답을 할 수 있게 되었다.

하지만 LLM이 사용자들의 다양한 입력값을 바탕으로 복잡한 판단을 수행하게 되면서 새로운 보안 위협도 발생하고 있다. LLM은 명령의 출처를 식별하지 못하고 사용자가 입력하는 프롬프트를 무조건 신뢰하는 특성이 있어 공격자에게 새로운 공격 수단으로 이용될 수 있다. 이러한 공격 수단 중 하나인 프롬프트 인젝션(Prompt Injection)은 악의적인 프롬프트 입력이나 외부 콘텐츠에 명령을 삽입하여 공격자가 의도한 대로 동작하게 만드는 보안 위협이다. 특히나 프롬프트 인젝션은 무궁무진하게 표현이 가능한 자연어를 기반으로 이루어진다. 이는 전통적인 시그니처 기반 필터링이나 키워드 차단 방식만으로는 한계가 있음을 시사하며, 이에 대한 체계적인 이해와 분석이 요구된다.

따라서 본 논문에서는 프롬프트 인젝션을 보안 위협으로써 정의하여 LLM 환경에서의 프롬프트 인젝션을 공격 사례 동향을 분석하고, 이러한 공격 기법에 관한 선행 연구를 참고하여 대응 방안을 제안하고자 한다.

2. 이론적 배경 및 선행 연구

본 장에서는 이해를 돕기 위해 보안에서의 공격 기법의 하나인 인젝션과 본 논문의 주제인 프롬프트 인젝션 공격의 개념을 소개한다.

2.1 인젝션(Injection)

인젝션은 주입, 삽입의 뜻을 가진 단어이다. 보안에서는 공격자가 신뢰할 수 없는 코드나 데이터를 시스템 및 프로그램에 주입하는 공격을 의미한다. 본래 프로그램은 사용자의 입력을 받아 처리할 때 입력된 값을 안전하게 다루고자 입력값 검증을 거치게 된다. 하지만 이런 입력값

검증이 제대로 되지 않는 경우 공격자가 프로그램에 명령어나 쿼리, 스크립트 등을 주입할 수 있게 된다. 이에 따라 프로그램은 본래 개발자가 의도한 바와 달리 데이터베이스를 비정상적으로 조작되며 결과값이 다르게 나오도록 유도할 수 있다.

아이디나 암호 창과 같은 입력창에 SQL 쿼리를 삽입하여 공격을 시도하는 SQL 인젝션과 권한이 없는 상태로 웹페이지에 악성 스크립트를 주입해 다른 사용자의 브라우저를 공격하는 XSS(Cross-Site Scripting)가 대표적인 예다.

2.2 LLM 환경에서의 프롬프트 인젝션

프롬프트는 본래 컴퓨터가 사용자의 명령을 받아들일 준비가 되었음을 모니터에 나타내는 상태를 의미한다. 나아가 명령어 입력을 위한 인터페이스라고도 볼 수 있는데, CLI(Command Line Interface)가 대표적 예시다.

현재의 프롬프트는 생성형 인공지능에 특정 작업을 수행할 수 있도록 요청하는 텍스트 기반의 명령이나 질문, 요청 등의 입력값 등을 의미한다. 사용자가 악의적인 명령어나 지시를 프롬프트에 삽입하여 모델의 기본 응답 로직을 우회하거나 변경하도록 유도할 수 있는데, 이러한 공격 방식이 프롬프트 인젝션이다.

생성형 인공지능은 코드처럼 정해진 명령어를 따르는 것이 아니라 프롬프트 안의 문맥을 자연어로 해석하여 행동을 결정한다. LLM은 수십억 개 이상의 파라미터를 바탕으로 한 방대한 양의 텍스트 데이터를 학습하여 주어진 문맥을 이해하고 자연스러운 텍스트를 생성하는 것이 특징을 지니고 있기 때문이다.

따라서 사용자가 작성하는 프롬프트의 내용과 순서, 어조, 의미적 구조가 매우 큰 영향을 미치게 되는데, 이러한 LLM의 구조적 허점을 노려 프롬프트 인젝션과 같은 공격이 가능해지게 된다.

2.3 선행 연구

OWASP에서 선정한 LLM 관련 보안 위협 리포트인 OWASP Top 10 for Large Language Model Applications에서는 프롬프트 인젝션을 LLM 기반 애플리케이션의 대표적 보안 취약점 중 첫 번째로 소개하였다. 이러한 주입식 공격의 영향을 최소화하고자 입력 값 검증 강화, 입출력 필터링 적용, 권한 제어 및 최소 권한 액세스 시행 등의 방안들을 제안하였다.

Yupei Liu(2024) 등은 프롬프트 인젝션 공격 기법 5가지와 방어 기법 10가지를 이용해 10가지의 LLM 모델을 테스트하였다. 하지만 사전 예방적 방어는 공격에 대한 완전 방어가 어렵고, 사후 탐지형 방어는 오탐률이 높다는 한계를 도출하였다. Jingwei Yi (2023) 등은 간접 프롬프트 인젝션 공격의 위험성을 분석하였다. 그 과정에서 프롬프트 학습을 통한 방어를 하는 블랙박스 방식과 역공학적 학습 기반 방어를 하는 화이트박스 방식의 개발 및 효과 검증을 하여 효과를 검증하였다. 하지만 블랙박스 방어 기법이 공격 성공률을 낮춰주더라도 간접 프롬프트 공격을 완벽하게 막는 것은 불가능하다는 한계점이 있었다.

Sizhe Chen(2024) 등은 프롬프트와 데이터를 명확히 분리하는 구조화 질의(StruQ) 방식을 제안하였다. 프론트엔드에서 명령어와 데이터를 별도의 채널로 구조화하고 이를 인식하도록 LLM을 미세 튜닝하는 방식으로 지시어와 데이터의 혼동을 방지해 프롬프트 인젝션 내성을 크게 향상하며 정상적인 기능의 손실 없이 보안성을 효과적으로 높일 수 있음을 실험적으로 입증하였다. 또한 Ruiyi Zhang(2025) 등은 새로운 방어 기법으로 혼합 인코딩(Mixture of Encodings)을 제시하였다. Base64를 포함한 다양한 문자 인코딩 방식을 조합하여 프롬프트 인젝션 공격의 효율성을 저하해 실질적인 적용 가능성을 보여주었다.

3. 사례 분석 및 대응 방안

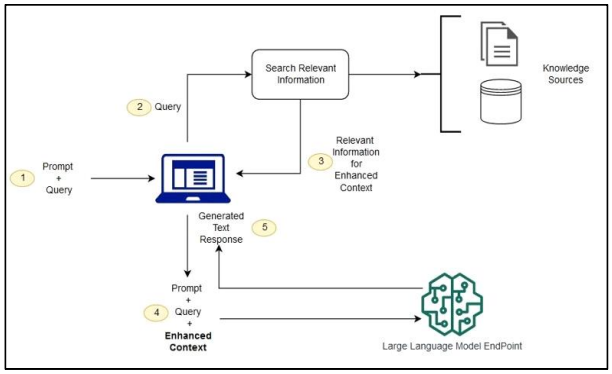
3.1 직접 프롬프트 인젝션

직접 프롬프트 인젝션 방식으로는 대표적으로 지침 무력화 공격, 역할 재할당 공격, 맥락 혼동 공격, 포매팅 악용 공격, 순차적 명령 공격, 코드 인젝션 공격 등 여섯 가지의 방법이 존재한다. 이러한 공격들은 일반적으로 기존의 언어모델 애플리케이션 프롬프트에 정상적인 텍스트를 입력하는 것이 아닌 악의적인 텍스트를 주입하며 언어모델이 비정상적인 행동을 수행하며 일으키는 보안 문제이다.[1]

실제 사례로 Microsoft의 Bing 검색 엔진에 통합된 Bing Chat(코드명 “Sydney”)은 2023년 초 일부 사용자에게 프리뷰로 공개되었는데, 출시 직후 프롬프트 인젝션을 통해 숨겨진 시스템 지침이 유출되는 사건이 벌어졌다.

3.2 간접 프롬프트 인젝션

간접 프롬프트 인젝션 공격은 직접 프롬프트 인젝션 공격과 달리 입력 프롬프트가 아닌 검색 증강 생성(RAG)을 통해 처리되는 콘텐츠 내에 내장되어 동작하는 경우를 말한다.[1]



[그림 1] RAG 사용 구조(AWS)

RAG에서는 검색 키워드가 많이 포함된 문서를 찾는 키워드 검색과 의미를 분석하여 벡터 간의 유사도를 계산하는 시맨틱 검색 두 방법을 결합한 하이브리드 검색 방법을 사용하는데, 이 키워드 검색에서 문제가 발생한다. 예를 들어 공격자가 악의적인 웹페이지를 만들어 두고 사이트 내에 키워드나 악의적인 지침, 코드를 삽입하면 사용자가 웹페이지를 LLM으로 사용 및 요약하는 과정에서 악의적인 지침이나 코드가 실행될 위험성이 있다.

실제로 2023년 3월 독일 CISPA 연구진들은 Microsoft Bing Chat에서 웹페이지 내 숨은 텍스트를 이용한 간접 프롬프트 인젝션 기법을 시연하는 사례가 존재한다. 해당 웹페이지를 요약하도록 요청할 때 챗봇은 악성 지시문을 읽고 역할을 변경하거나 개인정보를 요구하는 등의 취약점이 존재했다. 본 연구는 LLM이 외부 콘텐츠를 신뢰할 경우 발생할 수 있는 피싱 및 정보 탈취 위험성을 보여주었다.

또한 2023년 7월 보안 연구자 Johann Rehberger는 OpenAI의 ChatGPT 코드 인터프리터 기능에서 간접 프롬프트 인젝션을 통한 파일 유출 취약점을 발견하였다. 해당 공격은 악성 프롬프트가 포함된 웹페이지를 모델이 열람하도록 유도하여 그 내용을 코드 명령으로 오인하게 하여 /mnt/data 디렉토리에 위치한 사용자 업로드 파일의 내용을 인코딩 후 외부 공격자 서버로 전송하는 HTTP 요청을 생성하도록 하는 방식으로 수행되었다. 이는 사용자 입력이 아닌 외부 콘텐츠를 통해 모델의 실행 흐름이 조작될 수 있음을 보여준다. 또한 코드 실행 기능을 갖춘 LLM 환경에서 외부 입력에 대한 검증 부족이 직접적인 데이터 유출로 이어질 위험성을 보여준다.

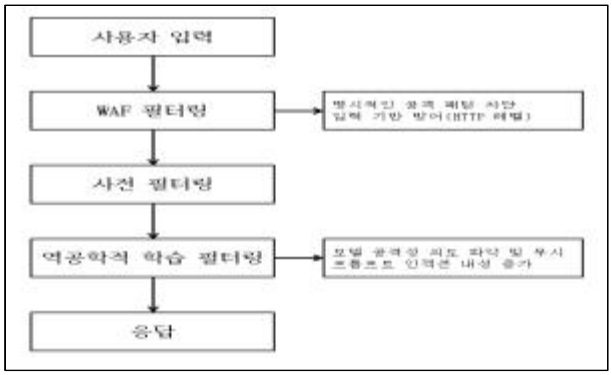
유형	방식	사용모델
직접방식	숨겨진 시스템 프롬포트 유출	Bing Chat
	플러그인 응답에 포함된 명령으로 다른 플러그인 실행	ChatGPT
간접방식	웹페이지 내 숨은 텍스트 이용	Bing Chat
	악성 프롬프트가 포함된 웹페이지를 모델이 열람 유도	ChatGPT
	악성 웹페이지의 지시문이 코드 인터프리터 통해 실행됨	ChatGPT
	Google Docs에 숨은 지시문 이미지 링크 통해 유출	Google Bard
	문서 또는 열람 파일 내에 시스템 명령 삽입	Claude

[표1] 프롬프트 인젝션 공격 사례 요약

3.3 대응 방안

상기된 선행 연구들 및 사례들로 LLM 환경에서의 프롬프트 인젝션 공격 대응 방안은 LLM에서 가장 중요한 해결 과제임을 가리키고 있다. 또한 LLM 환경에서의 프롬프트 인젝션을 방지하기 위해선 한가지 방식이 아닌 여러 방식을 통합한 보안 체계가 필요하다는 것을 도출할 수 있었다.

따라서 본 논문에서는 프롬프트 인젝션 공격의 대응 방안으로써 WAF(Web Application Firewall)와 역공학적 학습을 통한 프롬프트 인젝션 보안 솔루션을 제안하는 바이다. WAF와 역공학적 학습을 통합한 보안 솔루션은 다음과 같다.



[그림 2] 프롬프트 인젝션 보안 다이어그램

WAF 필터링을 통해 입력 기반 방어 및 명시적인 공격 패턴을 차단과 역공학적 학습으로 악의적 프롬프트 예제를 미리 학습시킨다면 이중 방어체계 구축으로 효과적인 결합 결과를 나타낼 것으로 예상한다. 이중 방어체계 구축으로 얻는 이점으로는 HTTP 레벨에서 일차적으로 걸러내며, 미처 걸러내지 못한 공격도 모델의 역공학적 학습으로 프롬프트 인젝션에 대한 대답을 무시할 가능성이 높아져 보안 방어력이 대폭 증가할 것이다.

하지만 WAF의 경우 학습 및 룰 기반 보안 정책이기에 새로운 변종 공격과 공격/비공격성 의도가 모호한 문장에는 탐지 실패 가능성이 존재한다. 또한 정상입력까지 공격으로 잘못 분류하여 모델이 작동을 거부하기 때문에 정상적인 사용이 어려워질 가능성 또한 존재하여 지속적인 역공학 학습데이터 및 룰 체계 업데이트가 필요하다는 한계점이 존재한다.

4, 결론

생성형 인공지능의 사용이 많아지며 LLM 환경에서의 보안 위협들 또한 대두되고 있다. 그중 직·간접 프롬프트 인젝션의 문제가 사회 여러 방면에서 이슈를 일으켰고, OWASP가 발표한 보안 위협 리포트에도 이름을 올리며 해결해야 할 보안 숙제로 남았다. 실제로 프롬프트 인젝션을 통해 BingChat의 숨겨진 시스템 지침(prompt)이 유출되거나 웹페이지 내 숨은 텍스트를 활용한 간접 프롬프트 인젝션 기법을 시연하는 등 여러 문제점이 밝혀졌다. 이는 직접적인 데이터 유출로 이어질 수 있음을 실증적으로 입증된 사례로 볼 수 있다. 본 논문을 통해 프롬프트 인젝션을 선행 연구한 자료들을 분석하여 역공학적 필터링 및 WAF를 이용한 프롬프트 인젝션 해결 방안을 제시하였다. 하지만 새로운 변종 공격과 공격/비공격성 의도를 알기 힘든 문장에는 탐지 실패 가능성이 존재하고, 정상입력까지 공격으로 분류하며 정상적인 사용이 어려워질 가능성 또한 존재해 지속적인 업데이트가 필요하다는 한계점이 있다. 이처럼 선행 연구와 본 논문에서 제안한 프롬프트 인젝션의 대응 방안들은 저마다 한계점이 명확하기에 추가적인 보완책과 지속적인 연구가 필요하다.

참고문헌

[1] 이상근, “LLM에 대한 프롬프트 인젝션 공격”, 한국정보처리학회 학술대회논문집, 제31권 2호, pp. 174-176, 10월, 2024년.

[2] OWASP, “LLM01: Prompt Injection,” OWASP Top 10 for LLM Applications, [온라인]. 이용 가능: https://genai.owasp.org/llmrisk/llm01-prompt-injection

[3] Ruiyi Zhang, David Sullivan, Kyle Jackson, Pengtao Xie, Mei Chen, “Defense against Prompt Injection Attacks via Mixture of Encodings”, arXiv preprint arXiv:2504.07467, 4월, 2025년.

[4] Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, Neil Zhenqiang Gong. “Formalizing and Benchmarking Prompt Injection Attacks and Defenses”, USENIX Security Symposium, 제33권, pp. 1831-1847, 8월, 2024년.

[5] Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, Fangzhao Wu, “Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models”, arXiv preprint arXiv:2312.14197, 12월, 2023년.

[6] Sizhe Chen, Julien Piet, Chawin Sitawarin, David Wagner, “StruQ: Defending Against Prompt Injection with Structured Queries”, arXiv preprint arXiv:2402.06363, 2월, 2024년.

[7] A. Obadiaru, "Prompt Injection Attacks: A New Frontier in Cybersecurity," Cobalt Blog, May 31, 2023. [온라인]. 이용 가능: https://www.cobalt.io/blog/prompt-injection-attacks.