거대 언어 모델 (Large Language Model, LLM)과 도구 결합의 보안성 연구

김주희 [1], 이병영 [2]
[1] 서울대학교 전기정보공학부 박사과정
[2] 서울대학교 전기정보공학부 교수

kimjuhi96@snu.ac.kr, byoungyoung@snu.ac.kr

Safety of Large Language Model-Tool Integration

Juhee Kim[1], Byoungyoung Lee [1]
[1]Dept. of Electrical and Computer Engineering, Seoul National University

요 약

이 연구는 거대한 언어 모델 (Large Language Model, LLM)과 도구를 결합한 시스템의 보안 문제를 다룬다. 프롬프트 주입과 같은 보안 취약점을 분석하고 이를 극복하기 위한 프롬프트 권한 분리 기법을 제안한다. 이를 통해 LLM-도구 결합 시스템에서의 사용자 데이터의 기밀성과 무결성을 보장한다.

1. 서론

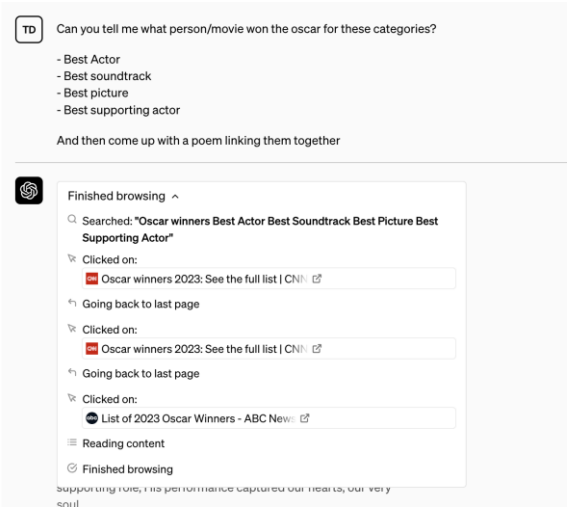

그림 1 ChatGPT 웹 브라우징 도구

GPT-4 [1], LLaMA [2], Gemini [3] 등 거대 언어 모델 (Large Language Model, LLM)의 높은 문제 해결 능력이 입증되며, LLM 과 결합한 어플리케이션이 생겨나고있다. 특히 ChatGPT [4], Bing Chat [5] 같은 챗봇 어플리케이션은 외부 도구 (tool) 과 다양한 결합을 시도해왔다.

LLM 결합 도구

첫째, 외부 데이터를 가져오는 도구는 모델이 학습한 정보 이상의 외부 정보를 활용하여 답변의 정확성을 높일 수 있다. 이를 통해 최신 데이터나 사용자에게 맞춤화된 정보를 답변에 반영할 수 있다. 예를 들어, 웹 브라우징, 데이터베이스 정보 검색, 개인 데이터 검색(이메일, 메시징 앱, 파일 시스템) 도구 등이 있다.

둘째, 외부에 영향을 주는 도구는 사용자가 허용한 동작에 대해 사용자의 권한으로 대신 동작을 취할 수 있다. 이는 챗봇의 답변 기능을 넘어 자동화 기능을 제공한다. 이메일 전송, 쉘 명령어 실행 등이 대표적인 예시이다.

LLM-도구 결합 시스템 구조

LLM-도구 결합 시스템 [6, 7, 8]에서 모델은 사용자의 입력에 적절한 도구를 사용할 수 있다. 구체적으로, 각 추론에서 모델은 중간 추론 (Thought) 을 반환하거나, 적절한 도구 (Tool)를 실행하여 관찰값 (Observation)을 얻거나, 최종 답변 (Final Answer) 을 제시한다. 시스템에 사용된 모든 데이터 (사용자 초기 입력, 중간 추론 결과, 관찰값 등)는 매 추론마다 모델에 입력으로 제공된다. 모델에 사용되는 모든 입력 데이터는 프롬프트 (Prompt) 라고 불리며, 가능한 많은 프롬프트를 사용하여 모델이 최선의 답변을 하도록 한다.

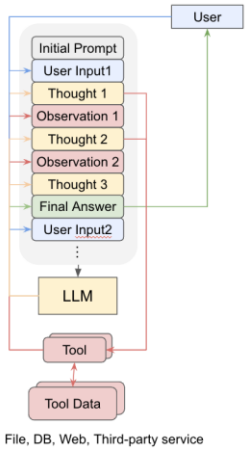

그림 2 LLM-도구 결합 시스템

2. LLM-도구 결합 시스템의 보안 문제

프롬프트 주입 (Prompt injection)
LLM-도구 결합 시스템에서 신뢰할 수 없는 외부 데이터를 가져오는 도구와 사용자 데이터에 접근하는 도구를 같이 사용할 경우, 외부 데이터가 시스템을 통제하여 사용자의 데이터를 유출하거나 변조할 수 있다. 이를테면 웹 브라우징 도구와 쉘 명령어 도구를 함께 사용할 경우, 신뢰할 수 없는 데이터를 유저의 명령으로 여겨 Remote code execution 공격 혹은 민감한 데이터 (e.g., /etc/passwd) 변조 및 유출이 가능하다 [9].

공격 모델
LLM 모델 및 LLM-도구 결합 시스템은 악의적인 목적이 없다고 가정한다. 여기서 피해자는 시스템의 사용자이며, 사용자는 개인 데이터와 외부 데이터를 모두 활용하여 개인화된 LLM-도구 결합 시스템을 사용하고 싶지만, 민감 데이터를 공격자에게 노출하거나 개인 데이터를 공격자가 위조하는 것을 원하지 않는다. 공격자는 특정 외부 데이터를 조작하여 도구의 관찰값을 바꿀 수 있으며, 악의적으로 LLM-도구 결합 시스템을 통해 이전에 권한이 없던 사용자 데이터에 접근하여 유출 또는 변조하는 것을 목표로 한다. 본 연구에서는 사용자 데이터의 기밀성 (Confidentiality) 과 무결성 (Integrity) 문제를 중점적으로 다루며, 가용성 (Availability) 문제는 다루지 않는다.

3. 프롬프트 권한 분리 (Prompt Privilege Separation)

본 연구에서는 각 프롬프트에 서로 다른 권한을 부여하여 LLM-도구 결합 시스템에서 사용자 데이터의 기밀성과 무결성을 보장하는, 프롬프트 권한 분리 기법을 제안한다.

권한이 높은 프롬프트를 추론에 사용할 경우, 모델이 모든 도구를 선택할 수 있도록 한다. 외부 데이터가 사용자 민감 데이터를 접근하는 것을 막기 위해, 권한이 낮은 프롬프트를 추론에 사용할 경우, 사용자의 개인 데이터를 접근하는 도구를 제외한다.

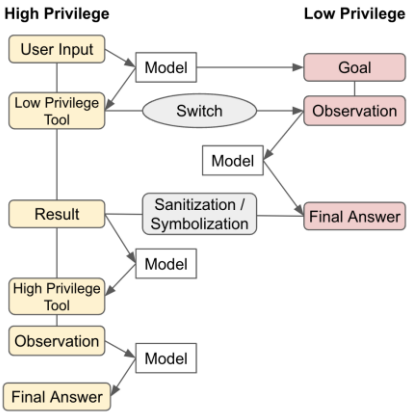

그림 3 프롬프트 권한 분리

프롬프트 권한
프롬프트의 권한은 데이터 생성지 (Source) 기준으로 정한다.

첫 번째, 사용자의 입력 값은 높은 권한을 가진다. 사용자는 개인의 데이터에 접근할 권한이 있으며, 시스템이 개인 데이터를 접근하는 것을 허용할 수 있다. 이는 사용자가 의도적으로 기밀성과 무결성을 해치는 데이터 흐름 또한 허용함을 의미한다. 이를 통해 시스템의 사용성을 보장할 수 있다.

두 번째, 도구가 불러온 개인 데이터 또한 높은 권한을 부여할 수 있다. 외부에서 영향을 줄 수 없는 개인 데이터는 안전하다고 여기고, 시스템의 사용성을 최대화하기 위해 높은 권한을 부여한다. 예를 들어, 사용자가 작성한 프로토콜 문서는 높은 권한을 부여한다.

세 번째, 도구가 불러온 신뢰할 수 없는 데이터는 낮은 권한을 부여하여 민감 데이터 접근을 방지한다. 예를 들어, 임의 공격자가 수정할 수 있는 데이터 (위키피디아, 웹 결과, 공유 데이터 등)는 낮은 권한을 가진다.

네 번째, LLM 중간 추론 결과는 이전 LLM 이 입력으로 사용한 프롬프트의 권한을 물려받는다.

세션 권한
기밀성과 무결성을 동시에 보장하기 위하여 높은 권한과 낮은 권한의 데이터는 별개의 세션에서 사용되어야한다. 낮은 권한 세션에서는 낮은 권한 프롬프트만 모델 인풋으로 사용되며, 공개 데이터를

읽고 쓰는 도구의 사용만 허용된다. 높은 권한 세션에서는 높은 권한 프롬프트만 모델 인풋으로 사용되며, 모든 도구 사용을 허용한다. 하지만 도구의 관찰값이 낮은 권한일 경우, 해당 데이터는 낮은 권한 세션에서 읽을 수 있다.

세션 전환 및 데이터 공유

시스템이 낮은 권한 데이터를 사용할 경우, 높은 권한과 낮은 권한 세션 전환이 필요하며, 이 때 보안성을 해치지 않도록 데이터가 공유되어야한다.

첫 번째, 높은 권한에서 낮은 권한으로 전환하는 경우는 낮은 권한의 관찰값이 발생했을 때이다. 이 때 관찰값과 함께 목적 (Goal) 이 낮은 세션에 제공되어야 한다. 목적은 모델이 생성하는 것으로, 사용자 기밀 데이터를 포함하지 않아야 한다.

예를 들어, 사용자 초기 입력이 “받은 메일함에서 가장 최신의 이메일을 읽고 요약” 일 때, 모델은 Mail 도구를 사용하여 최신 이메일을 불러온다. 이 때 이메일 발신자를 신뢰 못하는 경우, 이메일 데이터는 낮은 권한을 부여받는다. 시스템은 낮은 권한의 이메일 데이터와 “메일 요약하기" 라는 목표를 낮은 권한 세션으로 전달한다.

높은 권한에서 낮은 권한으로의 데이터 흐름에서, 높은 권한의 사용자 데이터가 낮은 권한 도구 데이터로 유출하여 기밀성 (Confidentiality) 을 해칠 수 있다. 본 연구에선 모델이 악의성이 없다고 가정하므로, 높은 확률로 모델이 불필요한 사용자 데이터 유출을 하지 않는다. 다만, 사용자의 명령이 의도적으로 높은 권한 데이터를 낮은 권한으로 전달하기를 암시했다면, 모델은 사용자의 명령대로 데이터를 전달할 수 있다. 모델의 hallucination 현상으로 판단을 잘못하는 것을 방지하기 위해 추가적으로 프롬프트 엔지니어링을 적용할 수 있다.

두 번째, 낮은 권한에서 높은 권한으로 전환하는 경우는, 낮은 권한 데이터의 처리 결과를 높은 세션으로 반환하는 때이다. 이 때, 낮은 권한 데이터가 높은 권한을 사용하는 것을 제한하기 위해 데이터의 악의성이 제거되어야한다. 이를 위해 다음 두 가지 방법을 사용할 수 있다.

1) Sanitization: 낮은 권한 세션의 데이터를 Sanitize 하거나, 일부 선택지 중 하나를 고르는 것으로 제한할 수 있다. 이로써 낮은 권한에서 높은 권한에 줄 수 있는 위험성을 최소화하되, 기능성은 최대화할 수 있다. 예를 들어, 이메일 요약 내용의 선택지로 “미팅 일정 관련”, “연구 관련”, “프로모션”, “공지” 등이 주어지고, 낮은 권한에서는 해당 선택지 중 하나만 반환할 수 있다.

2) Symbolization: 낮은 세션에서 반환할 데이터의 자유도를 더 높이되, 위험성을 최소화 하기 위해 낮은 세션 데이터를 Symbol 로 변환할 수 있다. 낮은 권한의 데이터는 symbol 로 높은 권한에서 참조될 수 있으나, 데이터 값은 symbol 에 캡슐화되어있어 높은 세션에 영향을 주지 못한다. 예를 들어, 낮은 권한 세션에서의 반환 값이 $L0 = {Sender: A 교수님, Message: LLM 보안 연구에서 참고할만한 최근 연구 논문 안내} 이라면, 높은 권한 세션에서는 “$L0.sender 로부터 $L0.Message 를 받았습니다.” 라고 표현할 수 있으며, 이는 사용자에게 “A 교수님으로부터 LLM 보안 연구에서 참고할 만한 최근 연구 논문 안내를 받았습니다.” 라고 보여진다.

낮은 권한에서 높은 권한으로의 데이터 흐름에서는 낮은 권한의 데이터가 높은 권한의 데이터, 이를테면 사용자의 명령을 악의적으로 수정하여 무결성 (Integrity) 을 해칠 수 있다. 하지만 앞서 제안된 Sanitization 과 Symbolization 기법을 활용하면 낮은 권한에서 높은 권한으로 흘러갈 수 있는 데이터의 범위를 제한하고 필요한 데이터만 전달하여,공격을 방지하되 사용성을 보장할 수 있다.

4. 향후 연구 계획

Prompt privilege separation 기법은 모델의 prompt privilege 및 privilege separation system 에 대한 이해를 요구한다. 이를테면, 높은 세션에서 낮은 세션으로 전환 시 목적을 정확히 설정해야한다. 낮은 세션에서 높은 세션으로 전환 시에는 Sanitization rule 을 안전하게 설정해야하고, Symbolization 의 경우 symbol 을 시스템에서 정한 문법으로 사용하여 파싱 문제가 생기지 않도록 해야한다.

이를 위해 다양한 사용 예시를 포함한 데이터 셋 구축이 필요하며, 모델의 시스템 이해도 향상을 위해 system prompt 제작, fine tuning 등이 필요하다.

더 나아가 3 가지 이상의 권한이 존재할 경우 (높은, 중간, 낮은 권한)에도, 본 연구의 프롬프트 권한 분리 모델을 확장하여 사용할 수 있을 것으로 보인다.

Acknowledgment

이 논문은 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2020-0-01840,스마트폰의 내부데이터 접근 및 보호 기술 분석)

참고문헌

[1] Achiam, Josh, et al. "Gpt-4 technical report." *arXiv preprint arXiv:2303.08774* (2023).
[2] Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." *arXiv preprint arXiv:2302.13971* (2023).
[3] Gemini, https://gemini.google.com/?hl=ko
[4] ChatGPT, https://chat.openai.com
[5] Bing Chat, https://www.microsoft.com/en-us/edge/features/bing-chat?form=MA13FJ
[6] Schick, Timo, et al. "Toolformer: Language models can teach themselves to use tools." Advances in Neural Information Processing Systems 36 (2024).
[7] Liang, Yaobo, et al. "Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis." arXiv preprint arXiv:2303.16434 (2023).
[8] Lu, Pan, et al. "Chameleon: Plug-and-play compositional reasoning with large language models." *Advances in Neural Information Processing Systems* 36 (2024).
[9] Greshake, Kai, et al. "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection." *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. 2023.