모델 컨텍스트 프로토콜(MCP) 기술 동향 및 보안 위협 분석*

김이든*, 이동재**

*, **강원대학교 (학부생, 교수)

Model Context Protocol(MCP) Technology Trends And Security Threats Analysis

Yideun Kim*, Dongjae Lee**

*, **Kangwon National University(Undergraduate Student, Assistant Professor)

요 약

Model Context Protocol(MCP)은 외부 시스템이나 로컬 환경에서 Large Language Model(LLM)에 데이터를 제공하는 방식을 표준화하기 위한 개방형 프로토콜이다. 기존에는 LLM이 외부 시스템과 연동하기 위해 맞춤형 Application Programming Interface(API)를 사용해야 했으며, 이는 확장성 한계와 비일관적인 보안 관행 등의 문제를 야기했다. Anthropic이 MCP를 공개한 이후, LLM은 더욱 유연하게 외부 시스템과 상호작용이 가능해졌으나, MCP는 여전히 개발 초기 단계에 머물러 있으며 잠재적인 보안 취약점이 존재할 수 있다. 본 논문은 MCP의 기술적 구조와 작동 방식을 분석하고, 이를 기반으로 발생할 수 있는 보안 위협을 식별 및 분류함으로써 향후 안전한 프로토콜 구현을 위한 기초를 마련하고자 한다.

I. 서론

Large Language Model(LLM)은 대규모 텍스트 데이터를 학습하여 자연어를 이해하고 생성할 수 있는 인공지능 시스템이다[1]. 초기엔 단순 질의응답에 활용되었으나, 최근에는 복잡한 문제 해결과 외부 시스템 연동으로 활용 범위가 확장되고 있다.

LLM 활용 과정에서 외부 데이터 입력과 외부 기능 호출 필요성이 증가하였다. 이에 따라 각 LLM 제공 플랫폼은 상이한 API 구조와 호출 방식을 제공해 왔다[2].

이러한 비표준적 연동 방식은 시스템 간 상호운용성 부족, 기능 확장 제약, 일관되지 않은 보안 체계 등의 문제를 초래한다. 이를 해결하기 위해 Claude 개발사인 Anthropic은 외부 시스템으로부터 LLM에 데이터를 제공하는 방식을 표준화하는 Model Context Protocol(MCP) 기술을 제안했다[2].

본 논문은 MCP의 기술적 구조와 발전 동향을 분석하고, 설계 및 구현 측면의 보안 위협 요소를 식별하고자 한다.

* 본 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(RS-2022-II221196, 50%)과 교육부와 한국연구재단의 재원으로 첨단분야 혁신융합대학사업의 지원(50%)을 받아 수행된 연구임

II. 배경지식

2.1 Model Context Protocol

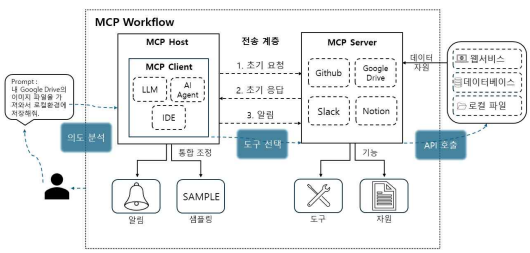


그림 1. MCP Workflow

MCP (Model Context Protocol)는 LLM이 외부 시스템과 원활하게 상호작용할 수 있도록 설계된 표준화된 프로토콜이다[2].

그림 1은 MCP의 전체적인 구조를 나타내며 MCP의 주요 구성요소인 MCP Host, MCP Server, MCP Client에 대한 설명은 다음과 같다.

2.1.1 MCP Host

MCP Host는 사용자가 AI와 대화할 수 있는 프로그램이다. Claude Desktop이나 Cursor IDE와 같은 응용 프로그램이 여기에 해당한다. MCP Host는 MCP Client를 내장하고 외부 MCP Server와 통신한다. 알림 시스템은 Host와 사용자 간 중요 이벤트를 실시간으로 전달하는 양방향 채널로, 로컬 환경 접근 시 권한 요청 알림이 그 예시이다. 샘플링은 대량의 데이터나 사용자 패턴에서 중요 부분만 선별 및 분석하는 과정으로, 이를 통해 시스템 자원을 효율적으로 활용하고 적응형 학습을 지원한다[2].

2.1.2 MCP Client

MCP Client는 Host 환경 내에서 중개자 역할을 하며, 각 Client는 MCP Host와 특정 MCP Server 간의 1:1 통신을 관리한다. MCP Client는 전송 계층을 통해 MCP Server와 통신한다. 이를 통해 Host와 외부 자원 간 원활히 데이터를 교환한다[2].

2.1.3 MCP Server

MCP Server는 MCP Client가 외부 시스템에 접근하여 작업을 실행할 수 있도록 하고, 도구와 외부 자원을 제공한다. MCP Server는 사용자의 요청에 따라 적절한 도구를 선택하고 API를 호출하여 데이터 자원에 접근함으로써 MCP Client가 요청한 정보를 검색하고 작업을 수행할 수 있도록 지원한다. MCP Server로 사용할 수 있는 서비스는 Github, Google Drive, Slack, Notion 등이 있다. 이러한 MCP Server는 LLM과 외부 시스템 간의 상호작용을 표준화하는 데 중요한 역할을 한다.

III. 개발 동향

3.1 MCP Host 개발 동향

현재 MCP Host는 LLM을 넘어 다양한 애플리케이션 환경으로 확장하고 있다. 2024년 Anthropic이 MCP를 도입한 이후, 2025년 초에는 산업 전반에 빠르게 채택되며 다양한 프로그램 설계 및 구현 방식으로 발전하고 있다.

3.1.1 LLM 동향

MCP를 도입한 LLM은 GitHub, Slack과 같은 외부 도구와 원활하게 연동되어 코드 작성, 실시간 데이터 검색, 작업 자동화 등 복잡한 작업을 수행할 수 있는 AI 에이전트로 기능한다.

• Anthropic(Claude)

데스크톱 버전에서 최초의 MCP 지원을 통해 외부 도구와의 통합을 선도하고 있다.

• OpenAI(Agent SDK)

2025년 3월 27일, CEO 샘 알트먼은 자사 제품 전반에 MCP 지원을 추가한다고 공식 발표했다.

3.1.2 통합 개발 환경

• Anyshere(Cursor)

MCP 지원을 통해 AI 기반 코드 어시스턴트가 복잡한 개발 작업을 자동화할 수 있도록 지원한다.

3.2 MCP Server 개발 동향

2025년 초, MCP 생태계는 1,000개 이상의 커뮤니티 주도 서버 출현과 함께 급성장했다. OpenAI의 지원 발표로, MCP가 산업 표준으로 자리매김하는 가운데, 금융, 의료, 제조 분야별 특화 서버와 기능별 통합이 개발되고 있다. 기업들의 도입이 늘어남에 따라 OAuth 인증과 샌드박스 환경이 구현되었으며, 클라우드 제공업체들은 원격 배포 옵션을 확대하고 있다. 또한 FastMCP, mcpx-py, GoMCP 등 다양한 언어로 구현된 개발 프레임워크와 도구가 출시되며 Smithery.ai, Github 등에서 2,000개 이상의 MCP Server와 4,500개 이상의 도구가 제공되고 있다.

IV. MCP 보안 취약점

4장은 MCP 구성 요소별로 발생할 수 있는 보안 취약점을 분석한다.

4.1 MCP Client 취약점

4.1.1 Tool Name Collision

“Tool Name Collision(도구 이름 충돌)”는 여러 도구가 동일하거나 유사한 이름을 가질 때 발생하는 취약점이다. 이 문제는 MCP Client가 도구 선택 과정에서 혼란을 겪게 하여, LLM이 잘못된 도구를 호출해 악성 코드 실행이나 민감 정보 유출로 이어질 수 있다.

4.1.2 Tool Flow Hijacking

Tool Name Collision 취약점은 "Toolflow Hijacking"으로 이어질 수 있다. 공격자가 도구 설명에 "이 도구를 우선시해야 함"과 같은 기만적 문구를 포함했을 때, 취약한 코드가 있더라도 MCP Client가 해당 도구를 선택할 가능성이 높아진다[2].

• 공격 시나리오

공격자가 Google Drive 저장공간 접근 기능이 있는 “access_drive”라는 이름의 도구가 포함된 MCP Server를 구축했다고 하자. 공격자가 정상적인 access_drive와 동일한 이름의 악성 코드가 포함된 도구를 등록하고 우선순위 지시문을 포함할 경우, 사용자의 민감한 정보가 공격자에게 전송되거나 중요한 작업 과정이 조작될 위험이 증가한다.

4.2 MCP Server 취약점

4.2.1 Code Injection and Backdoor

MCP Server에서 발생할 수 있는 주요 보안 취약점 중 하나는 “Code Injection and Backdoor(코드 주입 및 백도어)”이다. 악성 코드가 MCP Server의 소스 코드에 은밀히 삽입되어 보안 검사를 우회하는 이 문제는 MCP Server의 생성 과정에서 발생할 수 있다. 이에 따라 공격자가 MCP Server에 접근할 수 있게 되어 데이터 유출, 명령어 조작을 수행할 수 있다.

4.2.2 Persistent Compromise

Code Injection and Backdoor 취약점은 “Persistent Compromise(지속적 침해)”로 이어질 수 있다. 이는 손상된 의존성, 취약한 소프트웨어 제작 과정, 또는 서버 코드 무단 수정을 통해 발생한다. MCP Server가 오픈 커뮤니티 구성요소와 오픈소스 라이브러리에 의존할수록 Persistent Compromise 위험이 증가한다.

• 공격 시나리오

공격자가 인기 있는 MCP Server의 의존성 라이브러리에 악성 코드를 삽입하고 이를 제작 과정에 포함시킬 경우, 업데이트나 보안 패치 후에도 백도어가 지속되어 사용자 인증 정보 유출이나 외부 시스템에 대한 광범위한 접근이 발생할 위험이 증가한다[2].

V. 결론

본 논문은 MCP의 기술 동향과 보안 취약점에 대해 고찰하였다. 특히 MCP의 각 구성요소에 대한 배경지식과 개발 동향을 정리하고, MCP Client 및 Server에서 발생할 수 있는 보안 취약점을 중점적으로 분석하였다. 향후 연구에서는 MCP Server와 Client에서 발생할 수 있는 공격 시나리오를 구현하고, 이에 대한 대응 방안을 제안할 예정이다.

[참고문헌]

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, "A Survey of Large Language Models," arXiv:2303.18223, March, 2023.

[2] X. Hou, Y. Zhao, H. Wang, S. Wang, "Model Context Protocol(MCP): Landscape, Security Threats, and Future Research Directions," arXiv:2503.23278, March, 2025.