Agentic AI 기술을 활용한 사이버 공격 및 방어분야 최신 연구 동향

김지은*, 노현민*, 박찬규*, 조영호(교신저자)**
*국방대학교 국방관리대학원 국방과학학부 사이버컴퓨터공학과 석사과정
**국방대학교 국방관리대학원 국방과학학부 사이버컴퓨터공학과 교수
e-mail: younghocho@korea.kr

A Survey of Recent Research Trends in Cyber Attacks and Defense Using Agentic AI

Jieun Kim*, Hyeonmin No*, Changyu Park*, Youngho Cho(Corresponding Author)**
*Master's Course, Dept. of Cyber Security and Computer Engineering, Korea National Defense University
**Professor, Dept. of Cyber Security and Computer Engineering, Korea National Defense University

요 약

최근 Agentic AI는 목표를 이해하고 계획을 수립하며 외부 도구를 활용해 스스로 행동을 수행하는 형태로 빠르게 발전하고 있다. 이러한 기술은 사이버 보안 분야에서도 공격과 방어 양 측면에서 활용 가능성이 점차 확대되고 있다. 이에 본 논문에서는 Agentic AI의 구조와 이를 구성하는 핵심 기술인 RAG, ReAct, 멀티 에이전트 시스템(MAS)을 알아보고, 최근 사이버 공격 및 방어 분야의 연구 동향을 살펴보았다. 또한 공격 자동화, 취약점 탐지, 탈옥 방어, 통합 대응 체계와 같은 주요 연구 사례를 함께 분석하였다. 이를 통해 Agentic AI의 사이버 보안 적용 방향에 대한 이해를 높이고, 향후 관련 연구와 기술 발전 방향에 기초자료를 제공하고자 한다.

1. 서론

최근 인공지능은 목표를 이해하고 계획을 수립하며 스스로 행동을 수행하는 Agentic AI로 빠르게 발전하고 있다. 이러한 흐름은 대규모 언어모델(LLM)의 발전과 맞물려 더욱 가속화되고 있으며, Agentic AI의 핵심 기술로 활용되는 사례 또한 크게 증가하고 있다[1], [2]. 특히 최근에는 미국-이란 전쟁간 Palantir의 Maven과 같은 AI 기반 지휘·정보분석 지원 체계가 실제 작전 수행과 의사결정 보조에 활용되면서, Agentic AI 기술의 실질적 활용 가능성과 전략적 중요성이 더욱 부각되고 있다. 이러한 기술은 향후 군사·안보 분야를 포함한 다양한 영역에서 더욱 폭넓게 활용될 가능성이 높으며, 이에 따라 Agentic AI의 핵심 요소기술과 실제 적용 가능성에 대한 학술적·기술적 관심도 한층 높아지고 있다.

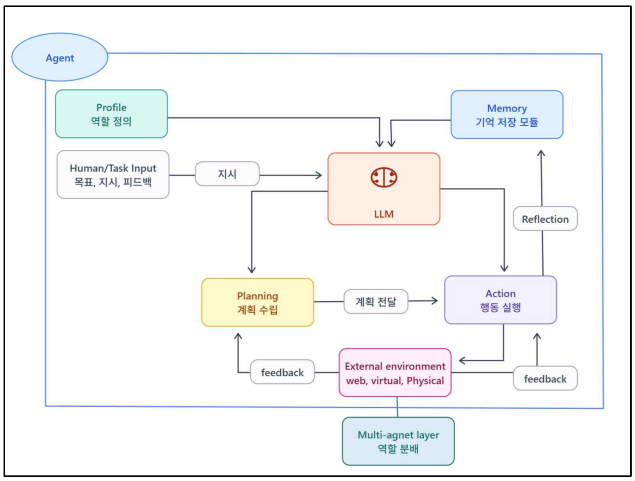
이러한 흐름에 따라 Agentic AI 기술은 사이버 보안 분야에서도 점차 중요한 연구 대상으로 부상하고 있다. 그러나 관련 연구는 아직 충분히 축적되었다고 보기 어려우며, 특히 사이버 보안에서 Agentic AI의 활용을 공격과 방어 측면으로 구분하여 분석한 기술 동향 연구는 더욱 부족한 상황이다. 따라서 관련 기술의 발전 흐름과 실제 적용 양상을 체계적으로 정리할 필요가 있다.

이에 본 논문은 Agentic AI 에 관한 최근 3개년간 주요 security 및 AI 학술지에 발표된 연구를 대상으로, 사이버 공격 및 방어 분야의 연구를 정리한다. 이를 바탕으로 Agentic AI의 사이버 보안 적용 양상을 최신 동향 중심으로 분석 및 정리함으로써 향후 관련 연구의 방향 설정과 기술적·학술적 논의의 기초 자료를 제공하고자 한다.

2. Agentic AI의 구조 및 핵심기술

Agentic AI는 대규모 언어 모델(LLM)의 고도의 추론 능력을 바탕으로 스스로 상황을 인식하고 계획하여 행동하여 목표를 추구하는 자율 에이전트(Autonomous Agent)이다 [2]. 그림 1은 단일 에이전트의 구조를 나타낸다. Agentic AI는 LLM을 중심으로 역할과 사용자가 제시한 목표 등을 통해 에이전트의 행동 방향을 결정하게 된다. 또한 목표 달성을 위해 스스로 계획을 수립하고, 이에 따라 행동을 수행하며, 그 결과에 대한 외부 환경의 피드백을 반영해 계획과 행동을 지속적으로 조정한

다. 특히, 피드백은 메모리에 저장되어 상황에 맞게 과거의 데이터가 활용되게 된다



[그림 1] Agentic AI 동작과정

이와 같은 Agentic AI의 자율적 의사결정과 행동 수행 능력을 효과적으로 구현하고 고도화하기 위해, 다양한 핵심 기술들이 함께 발전해 왔다.

먼저, RAG는 대규모 언어 모델(LLM)이 답변을 생성하기 전, 외부의 신뢰할 수 있는 데이터베이스나 문서에서 관련 정보를 검색하고, 검색한 정보 참고하여 답변의 정확성을 높이는 기술이다.

모델의 내부 파라미터에 저장된 지식에만 의존하는 기존 방식과 달리, RAG는 실시간으로 외부 데이터를 참조함으로써 정보의 정확성과 최신성을 획기적으로 높인다.

ReAct는 대규모 언어 모델(LLM)에서 별개로 다루어진 추론(Reasoning)과 행동(Acting)을 결합하여 문제를 해결하는 에이전트 설계 방식이다 [3].

이 방식은 피드백 루프를 통해 추가 학습 없이도 행동과 추론을 반복적하여 자체 교정을 가능하게 한다.

마지막으로, 멀티 에이전트 시스템(MAS)은 대규모 언어 모델(LLM) 기반의 다양한 지능형 에이전트들이 서로 소통하며 공통의 목표 달성을 위해 상호 작용하는 시스템이다. 각 에이전트에 역할 분담을 통해 구체적인 직책을 부여하고, 역할에 맞는 지식과 관점으로 문제 해결에 참여한다. 이들은 상호 검토와 의견 교환을 통해 오류를 실시간으로 식별·수정하여 답변을 생성한다 [4]. MAS은 단일 에이전트 방식보다 상호 보완을 통한 자율성, 높은 성공 확률에서 장점을 갖는다. 또한, 에이전트간에 소통을 통해 인공지능 환각과 같은 오류를 줄였으며, 높은 확장성과 자동화에 큰 의의를 갖는다.

3. Agentic AI 사이버 공격 연구 동향

3.1. HPTSA(Hierarchical Planning and Task-Specific Agents : 자율적 팀 기반 제로데이 공격)[5]

HPTSA는 팀 단위 LLM 에이전트에 계층적 구조(Hierarchical Structure)를 도입하여, 실전 제로데이(Zero-day) 취약점을 자율적으로 익스플로잇하는 프레임워크이다.

프레임워크 내에서 계획 에이전트(Planning Agent)는 전체 시스템을 탐색하고 공격 전략을 수립하며, 특정 보안 작업에 최적화된 하위 에이전트(Sub-agents)에게 구체적인 태스크를 할당한다. 할당을 받은 하위 에이전트는 SQL 주입이나 서버 설정 분석 등 특화된 임무를 수행하며, 그 결과를 다시 계획 에이전트에게 피드백하여 전체 공격 시나리오를 완성한다.

이러한 계층적 협업 모델은 LLM의 고질적 한계인 '장기적 계획(Long-term Planning)' 문제를 해결함으로써, 분류 체계상의 '악성 행위 수행' 위협이 지능화·극대화된 사례를 보여준다. 실험 결과, HPTSA는 사전 정보가 없는 실제 제로데이 취약점을 단일 에이전트 대비 약 4.3배 높은 성공률로 공략하며 에이전트 기반 공격의 실효성을 입증하였다.

3.2 PENTESTGPT(상태 인식 기반 자동 모의해킹)[6]

PENTESTGPT는 LLM의 한계인 '장기 문맥 망각'을 극복하기 위해 설계된 상태 인식 기반(State-aware) 자동화 프레임워크이다. 본 시스템은 인간 전문가의 사고 과정을 모방하여 추론, 생성, 파싱을 담당하는 세 개의 모듈을 통해 유기적인 침투 테스트를 수행한다.

핵심 동력은 트리 구조의 침투 테스트 매트릭스(PTM)이다. 추론 모듈은 PTM을 통해 공격 단계별 발견 사항을 관리하며 전체적인 맥락을 유지한다. 이를 바탕으로 생성 모듈은 Nmap, Metasploit 등 실제 도구의 명령어를 호출하고, 파싱 모듈은 실행 결과에서 핵심 데이터만을 추출하여 다음 의사결정을 지원한다.

작동 프로세스는 정보 수집 → 취약점 분석 → 익스플로잇 → 권한 상승으로 이어지는 체계적인 4단계를 따른다. 특히 PTM 구조를 통해 공격의 일관성을 확보함으로써 복잡한 시나리오에서도 중단 없는 자율 침투가 가능하다. 또한, 오류 발생 시 사용자와의 대화형 피드백을 통해 공격 경로를 유연하게 수정할 수 있어, 비전문가도 고도화된 모의해킹 업무를 수행할 수 있는 실무적 효용성을 제공한다.

3.3 MasterKey (시차 분석 기반의 보안 우회 자동

화)[7]

공격 에이전트가 방어 메커니즘을 역공학하여 우회 전략을 자율 학습하는 단계로 진화하고 있음을 보여주는 대표적 사례는 MasterKey이다.

MasterKey는 SQL 인젝션 탐지 기법을 응용하여 시차 분석(Time-based Analysis)을 도입하였다. 이는 질문 투고 후 첫 토큰 생성까지의 응답 지연 시간(Latency)을 정밀하게 측정하는 방식이다. 이를 통해 공격자는 비공개된 보안 필터가 특정 키워드에서 작동하는 패턴을 식별하고 내부 방어 기제를 역공학한다.

파악된 방어 체계를 무력화하기 위해 MasterKey는 성공적인 탈옥 데이터를 학습하여 미세 조정(Fine-tuning)된 전용 공격 LLM을 활용한다. 이 모델은 타겟 시스템의 최신 방어선을 우회하는 변종 프롬프트를 실시간으로 생성하며, 기존 방식 대비 공격 성공률을 약 21.58% 향상시켰다.

결과적으로 MasterKey는 보안 패치 이후에도 시차 분석을 통해 변화된 방어선을 즉각 재확인하고 공격을 재생성하는 실시간 교전 구조를 보여준다.

4. Agentic AI 사이버 방어 연구 동향

4.1 LLM×CPG (코드 속성 그래프 기반의 정밀 취약점 탐지)[8]

LLMxCPG는 LLM의 맥락 파악 한계로 인한 오탐(False Positive) 문제를 해결하기 위해 정적 분석 도구와 LLM을 결합한 프레임워크이다. 본 시스템은 코드의 제어·데이터 흐름과 구조 정보를 통합한 코드 속성 그래프(CPG)를 활용하여 보안 취약점을 정밀하게 식별한다.

분석 프로세스는 총 3단계로 진행된다. 먼저 LLM이 취약 가능 지점을 선별하면, 해당 지점 간의 연결성을 검증하기 위해 LLM이 스스로 CPG 질의문(CPGQL)을 생성 및 실행한다. 마지막으로 추출된 데이터 흐름 경로를 LLM이 재분석하여 최종 위협 여부를 판별한다.

이러한 방식은 정적 분석의 논리적 엄밀성과 LLM의 의미론적 추론을 결합하여 탐지 정확도를 획기적으로 높였다. 특히 단순 판별을 넘어 구체적인 실행 경로(Execution Path) 증거로 제시함으로써 분석의 설명 가능성을 확보하였으며, 여러 파일에 걸친 복잡한 취약점 경로를 효과적으로 추적할 수 있는 방어 체계를 구축했다는 점에서 중요한 학술적 의의를 가진다.

4.2 JBShield (은닉 표현 공간의 개념 조작을 통한 탈옥 방어)[9]

공격 에이전트가 가드레일을 지능적으로 우회함에 따라, 모델 내부의 연산 과정에서 유해성을 직접 탐지하고 차단하는 JBShield 프레임워크가 제안되었다.

JBShield는 고수준의 개념이 은닉 표현 공간에 인코딩된다는 점에 착안하여 두 가지 핵심 모듈을 운용한다. 먼저 탐지 모듈인 JBShield-D는 특이값 분해(SVD)를 통해 추출된 '유해 개념'과 '탈옥 개념'이 입력 프롬프트에서 동시에 활성화되는지를 검사한다. 이 방식은 단 30개의 샘플만으로도 높은 정확도로 공격을 식별하는 효율성을 보여준다.

이어지는 완화 모듈인 JBShield-M은 탐지된 공격의 은닉 표현을 직접 조작한다. 유해 개념은 강화하고 순응을 유도하는 탈옥 개념은 약화시킴으로써, 모델이 고정된 거부 메시지가 아닌 맥락에 맞는 안전한 응답을 스스로 생성하도록 유도한다.

이러한 방식은 추상적인 탈옥 메커니즘을 토큰 단위의 개념으로 매핑하여 뛰어난 해석력(Interpretability)을 제공한다. 또한 모델 전체를 재학습할 필요가 없어 경제적이며, 수동 설계나 최적화 기반 등 다양한 공격 유형에 대해 범용적인 방어 성능을 갖추었다.

4.3 PromSec & RACONTEUR (설계부터 사후 분석까지의 통합 방어 체계)[10][11]

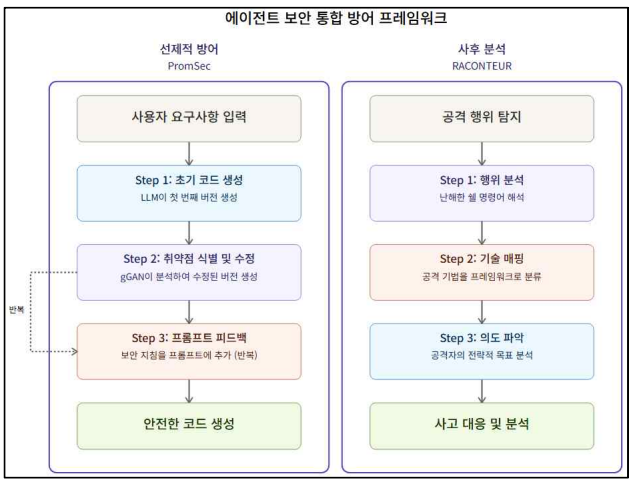
에이전트 시스템의 보안성을 완성하기 위해서는 코드 생성 단계의 선제적 방어와 침투 발생 시의 정밀한 사후 분석이 병행되어야 한다. 이를 위해 설계 단계의 보안을 담당하는 PromSec과 사후 대응을 지원하는 RACONTEUR를 결합한 포괄적 방어 프레임워크가 주목받고 있다.

먼저 PromSec은 에이전트가 코드를 생성하는 시점에 취약점 포함을 원천 차단한다. 취약점 수정 모델인 gGAN과 LLM을 상호작용 루프로 결합하여, 생성된 코드의 결함을 식별하고 이를 교정하기 위한 보안 지침을 프롬프트에 실시간으로 반영(Augmentation)한다. 이를 통해 에이전트는 기능적 정확성을 유지하면서도 다양한 소프트웨어 취약점(CWE)으로부터 안전한 결과물을 출력하게 된다.

이러한 선제적 방어에도 불구하고 발생할 수 있는 보안 사고에 대해서는 RACONTEUR가 보완적인 역할을 수행한다. RACONTEUR는 공격자가 실행한 난해한 쉘 명령어를 분석하기 위해 지식 증강 프롬프팅과 문서 검색기(Documentation Retriever)를 활용한다. 특히 분석된 행위를 MITRE ATT&CK 프레임워크의 전술 및 기술로 자동 매핑함으로써, 분석가가 공격자의 전략적 의도와 목

표를 즉각적으로 파악할 수 있도록 돕는다.

결과적으로 두 프레임워크의 결합은 에이전트의 동작 전반에 걸친 보안 가시성을 제공한다. PromSec이 '악성 행위 수행'의 가능성을 설계 단계에서 최소화한다면, RACONTEUR는 실제 위협 발생 시 이를 전문가 수준으로 역추적하여 사고 대응의 정확도를 높인다. 이러한 통합적 접근은 지능형 에이전트 환경에서 지속 가능한 보안 생태계를 구축하는 핵심적인 방안이 된다.



[그림 2] RACONTEUR 아키텍처

4. 결론

본 논문에서는 Agentic AI 및 AI Agent 기술을 활용한 사이버 공격 및 방어 연구의 최신 동향을 정리하였다. 이를 통해 Agentic AI가 목표 이해, 계획 수립, 도구 활용, 자율적 의사결정을 바탕으로 사이버 보안 분야에서 공격과 방어 양 측면에 모두 활용되고 있음을 확인하였다.

본 연구를 바탕으로 향후에는 Agentic AI의 핵심 요소 기술과 사이버 보안 적용 방안을 보다 체계적으로 연계하여 분석할 필요가 있다. 또한 관련 연구가 빠르게 확대되고 있는 만큼, 최신 기술 흐름과 실제 활용 양상을 지속적으로 정리하고 축적하는 후속 연구가 필요하다.

참고문헌

[1] Z. Xi et al., “The Rise and Potential of Large Language Model Based Agents: A Survey,” Science China Information Sciences, vol. 68, no. 2, Art. no. 121101, 2025.

[2] L. Wang et al., “A Survey on Large Language Model based Autonomous Agents,” Frontiers of Computer Science, 2025.

[3] S. Yao et al., “ReAct: Synergizing Reasoning and Acting in Language Models,” International Conference on Learning Representations (ICLR), 2023.

[4] C. Qian et al., “ChatDev: Communicative Agents for Software Development,” Proc. ICLR 2024, 2024.

[5] Y. Zhu, A. Kellermann, A. Gupta, P. Li, R. Fang, R. Bindu, and D. Kang, “Teams of LLM Agents can Exploit Zero-Day Vulnerabilities,” arXiv preprintarXiv:2406.01637, 2025.

[6] G. Deng et al., “PentestGPT: Evaluating and Harnessing Large Language Models for Automated Penetration Testing,” Proc. 33rd USENIX Security Symposium (USENIX Security 24), pp. 847-864, 2024.

[7] G. Deng et al., “MASTERKEY: Automated Jailbreaking of Large Language Model Chatbots,” Proc. NDSS 2024, 2024.

[8] A. Lekssays, H. Mouhcine, K. Tran, T. Yu, and I. Khalil, “LLMxCPG: Context-Aware Vulnerability Detection Through Code Property Graph-Guided Large Language Models,” Proc. 34th USENIX Security Symposium (USENIX Security 25), pp. 489-507, 2025.

[9] S. Zhang et al., “JBShield: Defending Large Language Models from Jailbreak Attacks through Activated Concept Analysis and Manipulation,” Proc. 34th USENIX Security Symposium (USENIX Security 25), 2025.

[10] M. Nazzal, I. Khalil, A. Khreishah, and N. Phan, “PromSec: Prompt Optimization for Secure Generation of Functional Source Code with Large Language Models (LLMs),” Proc. ACM CCS 2024, pp. 1-15, 2024.

[11] J. Deng et al., “RACONTEUR: A Knowledgeable, Insightful, and Portable LLM-Powered Shell Command Explainer,” Proc. NDSS 2025, 2025.