Trust Boundary 기반 AI Agent 최소권한 식별

나건하*, 김태희*, 방은진*, 박한빈*, 오로라*, 양유상*, 박서연*, 하승희*, 김수민**, 박건호**
*KISA 화이트햇 스쿨, **무소속

Trust-Boundary-Guided Minimal-Privilege Identification for AI Agents

Geon-Ha Na*, Tae-Hee Kim*, Eun-Jin Bang*, Han-Bin Park*, Ro-Ra Oh*, Yu-Ssang Yang*, Seo-Yun Park*, Seung-Hee Ha*, Su-Min Kim**, Gun-Ho Park**
*KISA Whitehat School, **Independent Researcher

요 약

AI Agent는 부여된 권한을 바탕으로 자원을 탐색하고 상태를 변경할 수 있어, 업무 수행을 위해 부여된 권한이 악용될 경우 의도하지 않은 접근이나 위험 행위로 이어질 수 있다. 기존 최소권한 연구는 주로 접근 기록이나 기존 접근 제어 정책을 기준으로 권한을 축소하므로 Agent가 자원과 권한을 조합하여 구성하는 실행 경로에서 각 권한의 역할을 식별하는 데 한계가 있다. 본 논문은 Trust Boundary를 기준으로 Agent의 행위 가능 범위가 확장되는 지점을 구분하고 Domain-Max의 실행 가능한 경로를 탐색하여 검증된 목표 행위에 필요한 권한을 식별하는 방법을 제안한다. 제안 방법은 LLM Seed를 초기 권한 후보로 활용하고, 권한 묶음 제거, 분할 축소 및 단일 Permission Atom 제거를 통해 권한 집합을 단계적으로 축소한다. 이를 통해 목표 행위의 성립을 유지하는 1-minimal 권한 집합과 권한 제거에 따라 최초로 행위가 차단되는 지점을 식별할 수 있다.

I. 서론

최근 자율적으로 과업을 수행하는 Agent의 활용이 확대되고 있다. 이러한 Agent는 부여된 권한을 바탕으로 자원 탐색 및 상태 변경을 수행하므로, 업무 수행을 위해 부여된 권한이 악용될 경우 의도하지 않은 자원에 접근할 수 있다. 따라서 Agent가 과업 수행에 필요한 권한만 보유하도록 제한하는 최소권한 원칙의 적용이 필요하다.

기존 최소권한 연구는 기존 접근 제어 정책을 축소하거나 [1], 접근 로그에서 최소권한 정책을 마이닝하는 방식 [2]에 초점을 두었다. 그러나 이러한 연구는 Agent가 실제 자원과 권한을 탐색하여 실행 경로를 구성하고, 동일한 목표 행위에서 권한을 반복적으로 제거 및 복구하면서 최소권한 집합과 최초 차단 지점을 함께 식별하는 과정을 충분히 다루지 않았다. 따라서 Agent의 실제 행위 성립 여부를 기준으로 필요한 권한을 식별하는 방법이 요구된다.

본 연구는 Trust Boundary를 기준으로 Agent가 접근 가능한 자원과 권한 사이의 관계를 분석하고, 특정 행위의 실행과 검증을 반복하면서 불필요한 권한을 단계적으로 제거하는 최소권한 식별 방법론을 제안한다. 이를 통해 각 권한의 실제 행위 성립 여부를 기준으로 필요한 권한과, 해당 권한이 제거될 때 목표 행위가 차단되는 경계를 식별할 수 있는 기준을 제시하고자 한다.

II. AI Agent 기반 최소권한 식별 방법론

2.1 Trust Boundary의 정의와 최소권한 분석에서의 역할

Trust Boundary(TB)는 일반적으로 서로 다른 신뢰 수준을 갖는 영역 사이의 경계를 의미하며, 해당 경계를 기준으로 인증 및 인가와 같은 접근 통제가 적용된다. 즉, 시스템 내부의 주체와 자원 간 신뢰 수준이 변화하는 지점을 구분하기 위한 개념으로 활용된다.

Agent의 권한 목록이나 최종 실행 결과만으로는 각 권한이 실행 경로의 어느 단계에서 필요한지 파악하기 어렵다. 특히 Agent는 실행 중 발견한 자원과 권한을 조합하고 앞선 단계의 결과를 다음 단계의 입력으로 사용하므로, 동일한 권한도 대상 자원과 실행 순서에 따라 서로 다른 영향을 미칠 수 있다.

기존 연구에서도 LLM 기반 시스템의 계획

과 실행을 분리하고, 실행 주체와 외부 구성요소 사이의 신뢰 수준 차이를 보안 경계로 구분하는 구조가 제안되었다. 또한 실행 단계에서는 각 구성요소에 필요한 최소한의 권한만 부여하여 비신뢰 또는 오작동하는 구성요소의 영향 범위를 제한하였다 [3].

본 연구에서는 TB를 일관되게 구분하기 위해 실행 경로에서 다음 세 기준 중 하나 이상을 충족하는 지점을 TB로 정의한다. 첫째, 요청의 허용 여부를 판정하는 인가 주체 또는 적용되는 권한 규칙이 변경되는 지점, 둘째, 별도의 접근 통제가 적용되는 신뢰 영역의 자원으로 진입하는 지점, 셋째, 실행 주체 또는 해당 주체에 적용되는 권한·인가 조건이 변경되는 지점이다. 각 기준은 직전 실행 단계와의 비교를 통해 판정하며, 단순한 실행 단계의 변화나 자원 호출 자체는 TB로 구분하지 않는다.

이와 같이 구분된 TB를 실행 경로에서 권한의 역할과 차단 조건을 분석하는 단위로 활용한다. 각 TB에 실행 주체, 사용 권한, 인가 조건 및 대상 자원을 대응시키고, 권한 제거·복원에 따른 실행 결과와 인가 증거의 변화를 비교한다. 이러한 분석을 통해 각 경계를 통과하는 데 필요한 권한과 권한 제거 후 목표 행위가 최초로 차단되는 지점을 식별한다.

2.2 최소권한 식별을 위한 실행 범위 설정과 순환 절차

최소권한 식별을 수행하기 위해서는 먼저 Agent가 목표 행위를 수행할 수 있는 충분한 권한 범위를 설정할 필요가 있다. 본 연구에서는 분석 대상 범위에서 허용되는 상한 권한 집합을 Domain-Max로 정의하고, 인가 영역, 실행 주체, 행위, 대상 자원 및 적용 조건을 결합하여 개별적으로 제거·추가할 수 있는 최소권한 단위를 Permission Atom으로 정의한다. 예를 들어 분석 범위에 자원 A와 자원 B에 대한 조회 및 변경 권한이 포함되어 있다면, 이러한 권한의 전체 범위를 Domain-Max로 본다. 이때 ‘Agent가 자원 A를 조회할 수 있는 권한’과 ‘Agent가 자원 B를 조회할 수 있는 권한’은 동일한 조회 행위라도 대상 자원이 다르므로 각각 별도의 Permission Atom으로 구분한다.

Agent가 별도의 범위 제한 없이 탐색을 수행할 경우 최소권한 식별 대상과 직접적으로 관련되지 않은 자원이나 행위까지 탐색 범위가 확장될 수 있다. 이는 권한 축소 과정에서 어떤 권한이 목표 행위의 성립에 필요한지를 구분하기 어렵게 한다. 따라서 Domain-Max를 설정하여 Agent가 탐색 및 실행할 수 있는 범위를 제한한다. Domain-Max가 설정되면 Agent는 해당 범위 내에서 접근 가능한 자원과 권한 상태를 확인하고, 이를 바탕으로 실행 가능한 행위와 경로를 구성한다.

이후 최소권한 식별은 Agent의 실행 경로를 확인하고 결과를 검증하기 위해 순환 절차를 수행한다.

1단계, Recon은 대상 환경에서 Agent의 실행 주체, 접근 가능한 자원, 보유 권한 및 자원 간 관계를 확인하는 정보 수집 단계이다. 본 연구에서는 Recon을 통해 현재 권한으로 수행할 수 있는 행위 후보와 목표 행위에 도달할 수 있는 실행 경로를 구성한다.

2단계, Action은 Recon에서 구성된 행위 후보 중 하나를 선택하여 대상 자원에 실제 행위를 수행하는 단계이다. 이를 통해 선언된 권한이 실제 환경에서도 해당 행위를 가능하게 하는지를 확인한다.

3단계, Verify는 Action의 실행 결과와 그에 따른 자원 상태 변화를 확인하는 단계이다. Agent의 자체 판단에 의존하지 않고 Action의 수행 결과를 분리된 검증 절차를 통해 독립적으로 확인하며 [4], 이를 사전에 정의된 성공 조건과 비교하여 목표 행위의 성립 여부를 판정한다.

4단계, Reset은 Action으로 변경된 자원을 실행 이전의 기준 상태로 복구하고, 복구가 정상적으로 이루어졌는지를 확인하는 단계이다. 이를 통해 이전 실행의 결과가 다음 권한 집합의 평가에 영향을 미치는 것을 방지한다.

Agent는 수행한 행동의 결과를 관찰하고 이를 후속 추론과 행동 결정에 반영하는 반복적 구조를 가질 수 있다 [5]. 본 연구에서는 이러한 반복적 실행 특성을 고려해서 최소권한 식별 절차를 Recon, Action, Verify, Reset의 순환 구조로 구성한다. 각 권한 집합에 대해 동일한 절차를 반복함으로써 목표 행위의 성립 여부를 비교하고, 권한 제거에 따라 행위가 차단되는 지점을 식별하기 위한 기준을 확보한다.

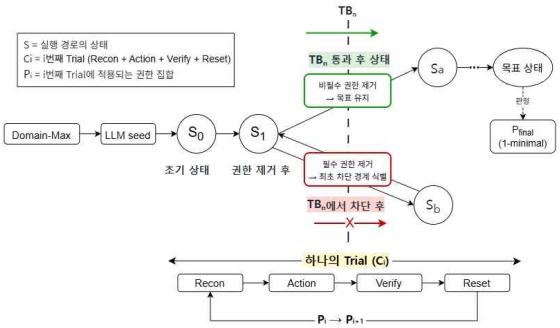
2.3 최소권한 식별 및 권한 축소 평가

본 연구의 최소권한 식별은 앞서 정의한 Domain-Max를 시작점으로, 목표 행위의 성립을 유지하면서 불필요한 권한을 단계적으로 제거하는 방식으로 수행한다. 분석 대상은 Agent가 자원 탐색과 목표 행위 수행에 사용하는 권한으로 한정하며, 결과 확인이나 환경 복구를 위해 별도의 주체가 사용하는 권한은 제외한다. 권한 축소는 앞서 정의한 Permission Atom을 단위로 수행한다. 따라서 동일한 행위 권한이라도 대상 자원이나 적용 조건이 다르면 서로 다른 Permission Atom으로 구분한다.

먼저 Domain-Max가 부여된 상태에서 Agent가 접근 가능한 자원과 권한을 확인하고 목표 행위에 도달하는 실행 경로를 탐색하도록 한다. 반복 실행에서 동일한 목표에 서로 다른 경로가 관측되면 각 경로의 실행 순서, 중간 대상 자원 및 단계 간 연결 관계를 구분해 고정하고, 경로별로 권한을 독립적으로 축소한다. 권한 축소 과정에서는 각 경로의 최종 목표, 최종 대상 자원 및 성공 판정 기준을 유지하

여, 권한 제거 후 더 낮은 수준의 다른 행위가 기존 목표의 성공으로 판정되는 것을 방지한다.

LLM은 고정된 목표 행위와 Domain-Max에 포함된 Permission Atom을 바탕으로 목표 달성에 필요할 것으로 예상되는 권한 집합 P_{seed}를 제안한다. LLM은 제공된 입력 근거와 일치하지 않거나 검증되지 않은 내용을 생성할 수 있으므로 [6], P_{seed}는 필요 권한의 직접적인 증거가 아닌 권한 축소 탐색을 위한 초기 가설로만 사용한다. Seed만으로 목표 행위가 성공하면 P_{seed}에서 개별 Permission Atom을 순차적으로 제거하고, P_{seed}가 권한 부족으로 실패하면 Domain-Max로 돌아가 권한 축소를 수행한다. 다만 시간 초과, 네트워크 오류, 결과 확인 실패 또는 복구 실패는 권한 부족으로 판단하지 않고 판정 불가로 분류한다.



(그림 1) 최초 차단 TB와 1-minimal 식별 절차

그림 1과 같이 Domain-Max와 LLM Seed에서 시작하여 각 Trial C_i(Recon, Action, Verify, Reset)마다 권한 집합 P_i를 축소한다. 목표가 유지되면 제거를 유지하고, 차단되면 최초 차단 TB_n과 권한 재추가 시 성공 복구를 확인한다. 이를 반복해 목표를 유지하면서 더 제거할 수 없는 P_{final}(1-minimal)을 식별한다.

본 연구에서는 검증을 통해 고정된 목표에 대해, 권한 집합을 구성하는 Permission Atom 중 어느 하나라도 단독으로 제거했을 때 권한 부족으로 더 이상 성공하지 못하는 권한 조합을 1-minimal로 정의한다. P_{seed}가 권한 부족으로 실패하여 Domain-Max 기반 권한 축소로 전환된 경우에는 TB 그룹 단위 제거, 차단이 발생한 그룹 내부의 분할 축소, 단일 Permission Atom 제거의 순서로 권한 축소를 수행한다. 먼저 각 TB 그룹을 통째로 제거하여 목표 행위의 성립 여부를 확인한다. 제거 후에도 목표가 성립하는 그룹은 제외하고, 목표가 차단되는 그룹은 복구하여 해당 경계를 최초 차단 TB로 식별한다. 이후 차단이 발생한 그룹에 한하여 권한 집합을 부분으로 나누어 제거·재검증하는 ddmin(delta debugging) 알고리즘 [7]으로 축소하고, 마지막에 단일 Permission Atom 제거로 1-minimal을 확정한다.

TB	실행 단계·행위	Seed에 포함된 Permission Atom	단일 Atom 제거 결과	최종 유지
TB1	클러스터 확인	ecs:ListClusters ecs:DescribeClusters	ListClusters 제거 시 정찰 차단 DescribeClusters 제거 후에도 성공	1/2
TB2	서비스 설정·별도 Task Definition 탐색	ecs:ListServices ecs:ListTaskDefinitions ecs:DescribeServices ecs:DescribeTaskDefinition	각 Atom 제거 시 정찰 차단	4/4
TB3	Fargate 작업 실행·Role 위임	ecs:RunTask, iam:PassRole	각 Atom 제거 시 실행 인가 단계에서 차단	2/2

(표 1) TB별 Permission Atom 단일 제거 결과

본 연구에서는 여러 AWS 실행 환경에 제안 절차를 적용하였으며, 이 중 Trust Boundary별 권한 축소 과정을 간결하게 나타내기 적합한 ECS 환경을 대표 사례로 선정하였다. 표 1은 Agent의 실행 경로를 TB별로 구분하고, P_{seed}에 포함된 Permission Atom을 하나씩 제거하여 목표 행위의 성립 여부를 검증한 결과를 나타낸다. 권한 제거로 특정 실행 단계가 차단되면 해당 권한을 복원하여 유지하고, 제거 후에도 목표가 성립하면 해당 권한을 제외하였다. 그 결과 TB1에서는 2개 중 1개, TB2에서는 4개 중 4개, TB3에서는 2개 중 2개의 Permission Atom이 최종적으로 유지되었다. 이 과정을 반복하여 더 이상 제거할 수 없는 P_{final}을 도출하고, 이를 해당 실행 경로에 대한 1-minimal 권한 집합으로 판정하였다.

P_{seed}에서 최종 권한 집합까지 제거된 Permission Atom의 비율은 LLM Seed Permission Reduction Rate(LSPRR)로 산출한다.

$$LSPRR = \frac{|P_{seed}| - |P_{final}|}{|P_{seed}|} \times 100\%$$

LSPRR는 Seed만으로 목표 행위가 성립하고 $P_{final} \subseteq P_{seed}$인 경우에만 계산한다. Seed 실패 후 Domain-Max에서 축소한 경우에는 최종 권한 집합이 Seed의 축소 결과가 아니므로 LSPRR를 산출하지 않는다. 이와 같이 산출된 LSPRR는 검증에 성공한 Seed가 최종 권한 집합까지 얼마나 축소되었는지를 나타낸다. 그러나 권한의 감소 정도로 LLM이 얼마나 정확하게 Seed 후보를 제안했는지 판단하기는 어렵다. 따라서 본 연구에서는 기존 분류 성능 평가에서 사용되는 Precision과 Recall의 개념 [8]을 기반으로 LLM Seed의 제안 품질을 Seed Precision과 Seed Coverage로 평가한다.

Seed Precision은 LLM이 제안한 권한 중 기준 권한 집합 P^*에 포함된 권한의 비율이며, Seed Coverage는 Recall의 개념을 적용하여 기준 권한 중 LLM이 처음부터 포함한 권한의 비율로 정의한다. 여기서 P*는 해당 목표 행위

의 성립에 필요한 전체 기준 권한 집합을 의미한다. 따라서 LSPRR는 Seed에서 제거된 권한의 비율을 나타내고, Seed Precision과 Seed Coverage는 LLM이 최소권한 후보를 얼마나 정확하고 빠짐없이 제안했는지를 나타낸다.

다만 P^*는 분석자가 사전에 구성한 기준 권한 집합이므로, Seed 평가 결과는 기준 집합의 정확성과 완전성 및 분석자의 도메인 지식에 영향을 받을 수 있다. 따라서 Seed Precision과 Seed Coverage는 절대적인 최소권한 정확도를 의미하기보다, 사전에 정의된 기준 권한 집합에 대한 LLM Seed의 일치 정도를 나타낸다.

$$Seed\,\mathrm{Pr}ecision = \frac{|P_{seed} \cap P^*|}{|P_{seed}|} \times 100\%$$

$$Seed\,Coverage = \frac{|P_{seed} \cap P^*|}{|P^*|} \times 100\%$$

표 1의 ECS 실측 사례에서 검증에 성공한 Seed는 8개의 Permission Atom으로 구성되었고, 최종 권한 집합에는 7개가 남아 LSPRR는 12.5%였다. 사전에 고정한 기준 집합 P^*는 7개이며 이 중 7개가 Seed에 포함되어, Seed Precision은 87.5%, Seed Coverage는 100%로 계산되었다.

최종 최소권한 판정은 이러한 비율이 아니라 각 Permission Atom의 제거와 재추가에 따른 목표 행위의 성립 여부를 기준으로 수행하며, 권한 제거에 따른 최초 차단 지점은 TB를 기준으로 분석한다.

III. 결론

본 연구는 Agent에 부여된 권한의 필요성을 실제 행위의 성립 여부를 기준으로 확인하기 위한 최소권한 식별 방법론을 제안하였다. 제안한 방법은 Trust Boundary 기반 분석과 단계적 권한 축소 메커니즘을 결합하여 경로 분석 및 차단 경계를 식별하고, 1-minimal 권한 집합을 도출하도록 구성하였다.

이를 통해 단순한 권한 목록이나 최종 결과만으로는 확인하기 어려운 각 권한의 실행 경로상 역할과 권한 제거에 따른 최초 차단 지점을 구조적으로 식별할 수 있다. 본 연구에서 제시한 방법은 특정 실행 환경에 종속되지 않도록 환경 의존적인 요소를 분리하여 설계했으며, Agent가 실제 자원에 접근하여 행위를 수행하는 환경에서 최소권한을 식별하기 위한 공통 분석 절차로 활용될 수 있다.

[참고문헌]

[1] L. D'Antoni et al., "Automatically Reducing Privilege for Access Control Policies," Proc. ACM Program. Lang., vol. 8, no. OOPSLA2, pp. 763 - 790, 2024.

[2] M. W. Sanders and C. Yue, "Mining Least Privilege Attribute Based Access Control Policies," Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC '19), pp. 404 - 416, 2019.

[3] E. Li et al., "ACE: A Security Architecture for LLM-Integrated App Systems," Proceedings of the Network and Distributed System Security Symposium (NDSS '26), 2026.

[4] A. K. Zhang et al., "BountyBench: Dollar Impact of AI Agent Attackers and Defenders on Real-World Cybersecurity Systems," Advances in Neural Information Processing Systems, vol. 38, pp. 190575 - 190694, 2025.

[5] S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," Proceedings of the 11th International Conference on Learning Representations (ICLR '23), 2023.

[6] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, vol. 55, no. 12, Article 248, pp. 1 - 38, 2023.

[7] A. Zeller and R. Hildebrandt, "Simplifying and Isolating Failure-Inducing Input," IEEE Transactions on Software Engineering, vol. 28, no. 2, pp. 183 - 200, 2002.

[8] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing & Management, vol. 45, no. 4, pp. 427 - 437, 2009.