AI 공격·방어 자동화와 보안표준의 재설계

최대선 숭실대학교 AI안전성연구센터장

1. 머리말

AI는 보안의 보조 도구에서 사이버 공간의 행위 능력을 확장하는 기반 기술로 이동하고 있다. 과거의 AI 활용은 악성코드 분류, 이상징후 탐지, 로그 요약, 보안 관제 보조처럼 방어자의 업무 효율을 높이는 방향에 가까웠다. 그러나 최근 LLM(거대언어모델, Large Language Model)과 에이전트 기술은 코드 이해, 취약점 후보 도출, 공격 경로 추론, PoC(개념검증, Proof of Concept) 작성, 패치 후보 생성, 사고 대응 절차 제안까지 수행할 수 있는 수준으로 발전하고 있다.

이 변화는 최근 공개된 앤트로픽(Anthropic)의 프로젝트 글래스윙(Project Glasswing)과 클로드 미토스프리뷰(Claude Mythos Preview) 사례를 통해 상징적으로 드러났다[1]. 해당 사례는 AI가 핵심 소프트웨어, 오픈소스 인프라의 취약점을 대규모 탐지할 수 있다는 사실을 드러냈으며, 동시에 같은 능력이 공격 코드 생성, 악용 가능성 검토에도 쓰일 수 있다는 우려를 낳았다. 이른바 미토스 쇼크의 본질은 특정 모델의 성능이 아니라 보안 활동의 속도, 규모, 비용 구조가 바뀌었다는 데 있다.

기존 보안표준은 주로 사람이 설계하고, 사람이 점검하고, 사람이 공격하는 세계를 전제로 발전했다. 조직은 통제 항목을 수립하고, 정해진 주기로 보안 점검을 수행하고, 취약점 목록과 패치 상태를 관리한다. 이러한 체계는 여전히 필요하지만 AI가 취약점 탐지, 공격 가능성 평가, 보안 조치 제안, 자동 대응에 개입하면 충분하지 않다. 표준은 더 이상 통제 항목의 존재 여부만 묻는 데 머물 수 없다. AI가 만든 판단과 산출물이 검증 가능한지, 공격 가능 정보가 적절히 통제되는지, 자동 대응이 안전장치를 갖췄는지, AI 에이전트의 행위가 감사 가능한지를 함께 다뤄야 한다.

2. AI 기반 공격·방어 자동화

2.1 취약점 발견과 악용 검증 자동화

AI 기반 자동화가 보안에 미치는 첫 번째 변화는 취약점 발견 비용의 하락이다. 기존 정적 분석 도구는 규칙, 패턴, 알려진 취약 함수, 특정 언어의 구문 구조에 의존하는 경우가 많았다. 반면 LLM은 소스 코드, API 문서, 설정 파일, 로그, 의존성 정보, 과거 취약점 패치 이력을 함께 해석하면서 코드가 어떤 의도로 작성됐고 외부 입력이 어떤 경로를 거쳐 위험 연산에 도달하는지를 추론할 수 있다. 공격자 관점에서는 공개 저장소, 패키지 의존성, 클라우드 설정, 웹 API, 컨테이너 이미지, 플러그인 생태계가 모두 자동 분석 대상이 된다.

두 번째 변화는 익스플로잇 검증 속도의 단축이다. AI는 취약점 후보를 찾는 데서 멈추지 않고

입력 조건을 추론하고, 실패 로그를 해석하며, 유사한 취약점 패턴을 비교하는 과정을 통해 공격 가능성을 빠르게 평가할 수 있다. 공개 취약점이 실제 공격으로 전환되는 시간은 이미 짧아지는 추세이고, AI는 이 전환 과정을 더 빠르게 만들 수 있다. 따라서 표준 관점에서 중요한 점은 실행 가능한 공격 절차나 코드 자체가 아니라, 그러한 정보가 '어떤 수준의 민감 정보를 구성하는지'이다.

취약점 보고서, 재현 절차, PoC, 패치 검증용 테스트는 방어에 필요하지만, 통제되지 않은 공개는 공격 자동화를 촉진한다. AI가 생성한 분석 결과는 '보안 점검 결과'와 '공격 가능 정보'라는 두 성격을 동시에 가진다. 표준은 이 둘을 구분해 취급해야 하며, 방어 검증에 필요한 상세 정보와 외부 공개가 가능한 요약 정보를 분리하는 기준을 마련해야 한다.

2.2 방어 운영 자동화와 이중용도성

AI는 공격자에게만 유리한 기술이 아니다. 방어자도 AI를 통해 취약점 후보를 자동 선별하고, 실제 노출된 자산과 연결해 우선순위를 정하며, 보안 경보를 상관 분석할 수 있다. 동시에 침해 흔적을 시간순으로 재구성하고, 대응 절차를 제안할 수도 있다. 특히 보안 인력이 부족한 조직에선 AI가 반복 분석 업무를 줄이고 전문가가 판단해야 할 고위험 사안에 집중하도록 도울 수 있다. 그러나 AI가 방어에 쓰인다고 해서 산출물이 곧바로 신뢰가능한 것은 아니다.

필자가 수행한 최근 연구[5]도 이러한 방어 자동화의 가능성을 보여준다. 해당 연구는 공개 취약점에서 반복되는 공격 경로를 구조화하고, 이를 LLM 프롬프트 규칙으로 전환해 취약점 후보를 찾으며, Source-to-Sink 흐름 추적과 오탐 필터링을 결합했다. 이는 AI가 보안 활동의 일부를 수행할 수 있음을 보여주는 동시에, AI 기반 방어 산출물이 표준화된 증거와 검증 절차 없이는 운영 현장에 안정적으로 편입되기 어렵다는 점을 시사한다.

결국 AI 기반 보안 자동화의 핵심 특징은 이중용도성이다. 취약점 탐지는 방어자의 사전 점검이 될 수 있지만 공격자의 목표 선별도 될 수 있다. PoC는 방어자의 재현성 검증 도구가 될 수 있지만 공격자의 무기화 재료도 될 수 있다. 패치 제안은 보안 개선에 기여할 수 있지만 우회 분석의 단서가 될 수도 있다. 따라서 표준은 AI 기능 자체를 선하거나 악한 것으로 분류해서는 안 된다. 더 중요한 것은 사용 맥락, 접근 권한, 산출물 민감도, 사람 승인 절차, 감사 가능성이다.

3. 기존 보안표준 전제의 균열

AI 기반 공격·방어 자동화는 기존 보안표준의 몇 가지 기본 전제를 흔든다.

첫째는 사람 중심 위협 모델이다. 기존 표준은 공격자와 방어자를 주로 사람, 조직 또는 자동화 도구의 운영자로 본다. 그러나 AI가 분석과 판단의 상당 부분을 수행하면 행위 주체가 복합화된다. 공격자는 AI에게 정찰과 취약점 후보 생성을 맡기고, 방어자는 AI에게 탐지와 우선순위화를 맡긴다. 이때 실제 행위의 책임은 모델, 도구, 운영자, 승인자, 조직 사이에 분산된다.

둘째는 정기 점검 중심 평가의 한계다. 많은 보안 점검과 인증은 정해진 시점의 상태를 평가한다. 그러나 AI가 취약점 발견과 악용 가능성 검토를 가속하면 평가 시점과 실제 위험 발생 시점 사이의 간극이 커진다. 모델 업데이트, 도구 추가, 권한 변경, 신규 플러그인 연동, 외부 데이터 소스 교체가 발생할 때마다 위험은 새롭게 형성된다. 연 1회 또는 반기 단위 점검은 더 이상 충분한

속도를 제공하지 못한다.

따라서 표준은 출시 전 평가와 함께 운영 중 지속 검증, 변경 발생 시 재평가, 고위험 산출물에 대한 즉시 검토 절차를 포함해야 한다. 또한 '누가 실행했는가'뿐만 아니라 '어떤 AI가 어떤 판단을 지원했고 누가 승인했는가'를 기록할 수 있어야 한다. 사람 중심 위협 모델은 사람-AI 협업 모델로 확장돼야 한다.

셋째는 제품 단위 인증의 한계다. AI 활용 보안 활동에서는 모델, 프롬프트, 에이전트 프레임워크, 코드 실행 환경, 클라우드 권한, 로그·티켓·배포 시스템이 결합돼 하나의 보안 활동을 구성한다. 따라서 특정 보안 도구가 인증을 받았는지 여부만으로는 전체 보안 활동의 신뢰성을 판단하기 어렵다.

넷째는 취약점 목록 중심 관리의 한계다. 기존 취약점 관리는 CVE(Common Vulnerabilities and Exposures) 식별자, CVSS(Common Vulnerability Scoring System) 점수, 패치 여부를 중심으로 한다.

이 방식은 알려진 취약점의 추적과 우선순위화에 효과적이다. 그러나 AI는 개별 취약점 목록보다 반복되는 공격 경로와 취약 구현 패턴을 더 빠르게 학습하고 활용할 수 있다.

필자가 수행한 연구[5]는 2022~2025년 공개 취약점 761건을 분석해 외부 입력 경유, 역직렬화 경유, 동적 코드 실행 경유라는 세 가지 반복 경로를 도출했다. 이 결과는 AI 기반 보안 자동화가 CVE 목록을 단순 조회하는 단계를 넘어 공격 경로 패턴을 활용하는 방향으로 이동할 수 있음을 보여준다. 표준은 제품 단위 평가와 취약점 목록 관리를 유지하되, 운영 프로세스, 공격 경로, 자산 맥락, 자동화 가능성을 함께 평가하는 구조로 확장돼야 한다.

4. AI 활용 보안 이슈

4.1 탐지 결과의 증거성과 재현성

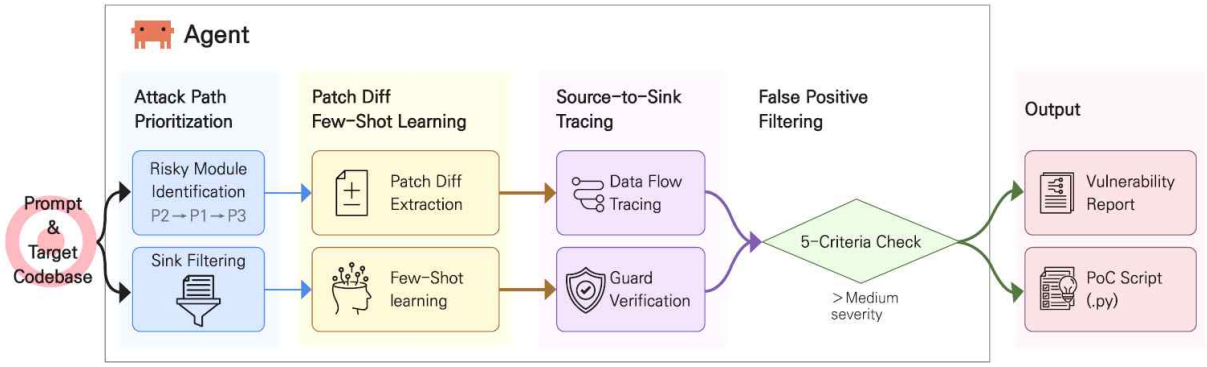
AI가 보안 활동에 활용될 때 가장 먼저 표준화돼야 할 대상은 탐지 결과의 증거성이다. AI가 취약점을 탐지했다는 주장을 할 때, 결과만으로는 충분하지 않다. 어떤 입력 지점에서 데이터가 시작됐는지, 어떤 모듈을 통과했는지, 어떤 위험 연산에 도달했는지, 공격자가 그 입력을 실제로 제어할 수 있는지, 기본 설정에서 재현되는지, 보호장치가 모든 분기에서 강제되는지까지 함께 제시돼야 한다. 이는 단순한 설명 가능성 요구가 아니라 취약점 보고의 책임성과도 연결된다.

두 번째 쟁점은 재현성과 검증 가능성이다. AI 판단의 가장 큰 위험은 그럴듯하지만 재현되지 않는 결과다. 보안 활동에서 재현성은 선택 사항이 아니다. 취약점 후보가 실제로 위험한지 확인하려면 버전 정보, 실행 환경, 입력 조건, 권한 조건, 보호장치 우회 여부, 로그, 예상 영향도, 기존 취약점과의 중복 여부가 정리돼야 한다. 근거가 불충분한 보고는 오픈소스 유지관리자와 공급자에게 불필요한 부담을 만들고, 반대로 실제 취약점을 발견했음에도 근거가 부족하면 조치가 지연된다.

필자가 수행한 연구[5]의 4단계 절차는 이러한 관점에서 참고할 수 있다. 공격 경로 우선순위화는 위험 모듈을 먼저 식별하고, 패치 차이 기반 예시 학습은 취약점 패턴을 구조화하며, Source-to-Sink 추적은 외부 입력에서 위험 연산까지의 흐름을 확인한다. 오탐 필터링은 공격자 제어 가능성, 기본 설정 트리거 가능성, 영향도, 중복 여부를 점검한다.

4.2 PoC·패치·자동 대응 산출물

[그림 1]은 LLM 기반 취약점 탐지 절차가 취약점 보고서와 PoC 스크립트까지 생성할 수 있음을 보여준다. 여기서 표준화의 초점은 AI가 취약점을 잘 찾는지에만 있지 않다. 공격 경로 우선순위화 단계의 기준, 패치 차이 학습에 사용되는 근거, Source-to-Sink 추적의 완전성, 오탐 필터링의 판단 기준, 보고서와 PoC의 분리 보관, 사람 검증 여부가 모두 표준화 대상이 된다. 특히 PoC는 방어자에게 필요한 검증 자료이지만 공격자에게는 악용 정보를 제공할 수 있으므로, 공개용 요약본과 검증용 상세본을 구분하고 접근 권한을 통제하는 기준이 필요하다.



출처: 류석준, 박소희, 옥도민, 최대선, 'AI/ML 프레임워크 취약점 공격 경로 분석 및 LLM 프롬프팅 기반 경량 버그헌팅 방법론', 한국정보보호학회, 2026[5].

[그림 1] AI 기반 취약점 탐지 절차와 표준화 대상 산출물

AI 생성 패치와 코드 검토도 중요한 쟁점이다. AI는 취약점 패치도 제안할 수 있지만, AI가 만든 패치는 곧바로 배포할 수 있는 정답이 아니라 검토 대상이다. 취약점을 제거했는지, 기능 회귀를 만들지 않았는지, 새로운 취약점을 유입하지 않았는지, 의존성이 바뀌었는지, 라이선스 위험이 있는지 확인해야 한다. 특히 중요 시스템에서는 AI가 생성한 코드, 사람이 수정한 코드, 자동 테스트 결과, 승인자, 배포 이력을 함께 남겨야 한다.

보안 관제 영역에서 AI는 경보를 분류하고 차단 정책을 제안하며, 계정 잠금, 네트워크 격리, 키 폐기, 접근 권한 회수 같은 조치를 자동화할 수 있다. 그러나 자동 대응은 오탐일 때 업무 중단을 초래한다. 표준은 자동 대응을 위험 등급별로 구분하고, 고위험 조치에는 사람 승인을 요구하며, 긴급 중지 기능과 롤백 절차를 포함해야 한다. AI가 방어 속도를 높이는 만큼 잘못된 방어 조치의 피해도 커질 수 있기 때문이다.

4.3 신원·권한·감사 로그

AI 에이전트가 보안 도구, 형상관리 저장소, 클라우드 콘솔, 배포 파이프라인에 접근하면 핵심 쟁점은 모델 성능이 아니라 행위 통제가 된다. 어떤 모델이 어떤 버전으로 판단했는지, 어떤 도구를 호출했는지, 어떤 권한으로 실행했는지, 누가 승인했는지, 결과가 어떻게 반영됐는지 추적할 수 있어야 한다. AI 에이전트에 사람이나 서비스 계정과 구분되는 신원, 작업 단위 임시 권한, 고위험 작업 승인, 도구 호출 로그, 실행 결과 감사, 철회와 롤백 기준이 필요하다.

이 지점에서 디지털 신원과 감사 가능성은 별도의 부가 기능이 아니라 AI 기반 보안 활동의 핵심 통제 항목이 된다. 기존 로그가 사람의 로그인, 시스템 이벤트, 네트워크 흐름을 기록하는 데 초점을 뒀다면, AI 시대의 로그는 모델 판단, 프롬프트, 입력 데이터, 도구 호출, 권한 부여, 사람 승인, 실행 결과, 롤백 여부를 하나의 감사 사슬로 연결해야 한다. 이 감사 사슬이 없으면 사고가 발생했을 때 AI가 제안했는지, 사람이 승인했는지, 자동화 시스템이 실행됐는지 구분하기 어렵다.
또한 AI 에이전트의 권한은 장기 고정 권한이 아니라 작업 단위 임시 권한이어야 한다. 취약점 점검, 패치 후보 작성, 티켓 생성, 테스트 실행, 배포 승인 요청은 서로 다른 위험 등급을 갖는다. 표준은 작업별 권한 범위, 승인 조건, 로그 보존 기간, 비정상 행위 탐지, 권한 철회 기준을 구분해야 한다. AI 에이전트가 방어자의 업무를 대신할수록, 그 행위는 더 강하게 기록되고 더 쉽게 되돌릴 수 있어야 한다.

5. 보안표준의 개정 방향과 국내 과제

5.1 지속 검증과 위험 맥락

AI 공격·방어 자동화 환경에서 보안표준의 첫 번째 수정 방향은 정적 통제에서 지속 검증으로의 전환이다. 기존 표준은 통제 항목의 존재 여부와 점검 시점의 적합성을 확인해 왔다. 그러나 AI 보안 도구, 모델 버전, 프롬프트, 데이터 소스, 권한, 자동 대응 정책은 운영 중 계속 바뀐다. 보안 도구가 업데이트되거나 에이전트에 새로운 도구 호출 권한이 부여되면 기존 평가 결과가 그대로 유효하다고 보기 어렵다. 따라서 변경 관리와 재평가 기준은 AI 시대 보안표준의 기본 요소가 돼야 한다.
즉 보안표준은 위험 산정도 취약점 등급 중심에서 악용 가능성 맥락 중심으로 넓어져야 한다. CVSS 점수와 패치 여부는 여전히 중요하지만, AI 시대에는 공격 자동화 가능성, 실제 인터넷 노출 여부, 자산 중요도, 권한 조건, 우회 가능성, 패치 난이도, 업무 영향을 함께 봐야 한다. 같은 취약점이라도 AI가 쉽게 공격 경로를 구성할 수 있는 환경에서는 위험이 더 크다. 표준은 점수 하나로 위험을 고정하기보다 자산 맥락과 자동화 가능성을 반영하는 평가 언어를 마련해야 한다.

5.2 증거 패키지와 보고 형식

AI 탐지 결과는 결론 하나가 아니라 증거 패키지로 관리돼야 한다. 취약점 후보, 입력 지점, 위험 연산, Source-to-Sink 경로, 재현 조건, 영향 범위, 오탐 필터링 결과, 사람 검증 결과, 공개 가능 범위가 함께 제시돼야 한다. 필자가 수행한 연구[5]에서 사용한 Source-to-Sink 추적과 오탐 필터링 절차는 AI 기반 방어 산출물을 검증 가능한 형식으로 바꾸는 사례다. 여기서 중요한 점은 연구 대상이 AI/ML 프레임워크였다는 사실이 아니라, AI로 수행한 보안 점검 결과가 근거와 재현 조건을 함께 가져야 한다는 점이다.
보고 형식은 여러 주체가 같은 의미로 읽을 수 있어야 한다. 개발자, 보안 관제 담당자, 법무·컴플라이언스 담당자, 외부 공급자, 오픈소스 유지관리자는 서로 다른 관심사를 가진다. 표준은 이들이 공유할 최소 필드를 정의해야 한다. 내부 요약, 취약점 후보, 재현 절차, PoC, 패치 후보, 자동 차단 명령, 외부 공개 문서는 위험도가 다르므로 저장 위치, 접근 권한, 승인자, 공개 범위도 달라야 한다.

5.3 고위험 산출물과 공개 절차

AI 생성 PoC와 공격 가능 정보는 방어 검증에 필요하지만 동시에 공격 자동화의 재료가 될 수 있다. 따라서 공개용 요약본과 검증용 상세본을 구분하고, 취약점 공개 전에는 접근 권한, 보관 기간, 복제 가능성을 제한해야 한다. 실제 위험이 확인되지 않은 후보가 외부로 유출되지 않도록 중복 검토와 오탐 필터링 기준도 필요하다. AI가 대량의 취약점 후보를 만들 수 있는 만큼, 공개 절차는 속도와 통제의 균형을 전제로 설계돼야 한다.

AI 생성 패치와 레드팀 평가 결과도 통제 대상이다. AI가 패치를 제안하면 취약점 제거 여부뿐 아니라 기능 회귀, 새로운 취약점 유입, 의존성 변화, 라이선스 위험, 배포 승인 이력을 함께 확인해야 한다. 벤치마크와 레드팀 평가셋은 공개성과 통제성 사이의 균형이 필요하다. 공격 코드 생성 가능성, 안전장치 우회 가능성, 탐지 정확도는 비교 가능한 형식으로 남기되 공개 범위는 단계화해야 한다.

조정된 취약점 공개 절차도 AI 환경에 맞게 보완돼야 한다. AI가 취약점 후보를 대량 생성하면 공급자는 검증 부담을 크게 느낄 수 있고, 실제 고위험 취약점이 대량 제보 속에 묻힐 수 있다. 표준은 우선순위 산정, 유효성 확인, 공급자 통보 기한, 공개 유예 기준, 패치 전 PoC 취급 기준을 제시해야 한다. 이는 연구자, 공급자, 사용자 모두를 보호하는 장치다.

5.4 운영 책임과 현장 적용

국내표준화는 선언보다 현장 점검 항목을 중심으로 해야 한다. 조직은 AI가 취약점 스캔, 코드 리뷰, 경보 분석, 티켓 작성, 패치 제안, 자동 차단 중 어디까지 관여하는지 먼저 목록화해야 한다. 이후 산출물 위험 등급을 정하고, 고위험 산출물에 대해 저장 위치, 접근 권한, 검토자, 승인자, 공개 범위를 정해야 한다. AI 에이전트가 실제 조치를 수행하는 경우에는 모델 버전, 도구 호출, 권한 부여, 승인자, 실행 결과, 롤백 가능성을 감사 로그로 남겨야 한다.

책임체계도 명확해야 한다. 모델 제공자는 모델의 능력과 제한, 안전장치, 업데이트 이력을 설명해야 하고, 보안 도구 제공자는 탐지 근거와 오탐 관리 방식을 제시해야 한다. 사용 조직은 AI 산출물의 승인, 배포, 공개 범위를 통제하고, 개발자는 AI 생성 패치가 보안 요구사항과 기능 요구사항을 모두 충족하는지 검토해야 한다. 이런 역할 분담이 없으면 사고 발생 시 "AI가 그렇게 판단했다"는 말만 남고 책임 주체가 사라진다.

현장 적용은 네 가지 시나리오에서 우선 시작할 수 있다. 첫째, AI 기반 취약점 점검은 대상 버전, 분석 도구, 탐지 기준, 재현 조건, 사람 검증 여부를 요구한다. 둘째, AI 기반 보안 관제는 사용 로그 범위, 판단 근거, 신뢰도 표시, 자동 조치 범위, 롤백 절차를 요구한다. 셋째, AI 생성 패치는 코드 출처, 테스트 결과, 승인자, 배포 이력을 남긴다. 넷째, AI 에이전트 보안 업무는 모델 버전, 도구 호출, 작업 단위 권한, 고위험 작업 승인, 실행 후 감사를 기본 요구사항으로 둔다.

5.5 국내표준화 우선순위

국내표준화의 우선순위는 AI 모델 자체의 안전성 평가와 AI 활용 보안 활동의 운영 기준을 구분하는 데서 출발해야 한다. 전자는 모델 위험 관리와 관련되지만, 후자는 취약점 탐지, 코드 검토,

관제 분석, 자동 대응, 취약점 공개 과정에서 생성되는 판단과 행위를 다룬다. 본 원고의 초점은 후자다. 따라서 우선 표준화할 항목은 AI 기반 취약점 탐지 결과의 공통 보고 형식, 고위험 산출물의 접근 통제와 공개 기준, AI 에이전트의 신원·권한·도구 호출·승인 로그, 사람 검증과 자동 처리의 경계다.

<표 1>의 핵심은 AI 시대 보안표준의 초점이 도구의 성능 평가에만 있지 않고, 보안 활동 과정에서 생성되는 산출물의 검증 가능성과 책임성 확보에 있다는 점이다. AI가 취약점을 탐지하고 패치를 제안하며 대응을 자동화할수록, 각 단계의 근거와 승인 절차, 감사 가능성을 함께 관리해야 한다.

<표 1> AI 기반 공격·방어 자동화 환경에서의 보안표준 보완 방향

구분	기존 관점	AI 환경 보완 방향	주요 산출물
취약점 탐지	취약점 존재 여부 확인	탐지 근거, Source-to-Sink 경로, 재현 조건 요구	증거 패키지
PoC 관리	재현 자료 중심	공개 범위와 접근 권한 분리	공개용 요약본, 검증용 상세본
AI 생성 패치	패치 적용 여부 확인	기능 회귀, 신규 취약점, 라이선스 위험 검토	리뷰 기록, 테스트 결과
자동 대응	대응 결과 확인	승인 기준, 긴급 중지, 롤백 절차	조치 로그
AI 에이전트	계정 관리 중심	신원, 권한, 도구 호출, 승인 기록	감사 로그

출처: 필자 작성

이 우선순위는 새로운 인증 항목을 무조건 늘리자는 의미가 아니라, 기존 보안 점검체계에 AI 산출물의 증거성, 접근 통제, 승인 로그를 결합하자는 의미다.

6. 맺음말

AI는 공격과 방어의 속도, 규모, 비용 구조를 동시에 바꾸고 있다. 공격자는 AI로 취약점 발견과 악용 가능성 검토를 가속할 수 있고, 방어자는 AI로 더 넓은 공격면을 더 빠르게 점검할 수 있다. 이 변화는 보안표준의 질문을 바꾼다. 이제 표준은 "통제가 존재하는가"뿐 아니라 "AI가 만든 판단과 행동을 누가 어떤 근거로 검증하고 책임지는가"를 물어야 한다.

필자가 수행한 연구[5]는 AI가 취약점 탐지와 오탐 필터링 같은 방어 활동에 실제로 활용될 수 있음을 보여준다. 그러나 그 자체가 본 원고의 결론은 아니다. 더 중요한 결론은 AI 기반 보안 활동이 현실화될수록 탐지 정확도보다 검증 가능성, 자동화 수준보다 책임 가능성, 도구 성능보다 운영 절차의 신뢰성이 중요해진다는 점이다.

따라서 AI 시대 보안표준은 지속 검증, 증거 패키지, 고위험 산출물 통제, AI 생성 패치 검토, 조정된 취약점 공개 절차를 포괄하는 동적 신뢰체계로 진화해야 한다. 창과 방패가 모두 AI가 되는 환경에서 표준의 역할은 AI 활용을 제한하는 것이 아니라, AI가 수행한 보안 행위가 검증 가능한 증거와 책임 있는 절차 안에서 이뤄지도록 만드는 데 있다.

[참고문헌]
[1] Anthropic, "Project Glasswing: Securing critical software for the AI era," 2026. Available: https://www.anthropic.com/glasswing
[2] NIST, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," 2023. Available: https://www.nist.gov/itl/ai-risk-management-framework
[3] OWASP, "OWASP Top 10 for Large Language Model Applications," 2025. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications/
[4] ISO/IEC, "ISO/IEC 27001:2022 Information security management systems - Requirements," 2022.
[5] 류석준, 박소희, 옥도민, 최대선, "AI/ML 프레임워크 취약점 공격 경로 분석 및 LLM 프롬프팅 기반 경량 버그헌팅 방법론," 한국정보보호학회, 2026.

[주요 용어 풀이]
- PoC(개념검증, Proof of Concept): 취약점이 실제로 재현 가능함을 보이기 위한 코드나 절차
- Source-to-Sink 추적: 외부 입력이 위험 연산에 도달하는 데이터 흐름을 추적하는 분석 방식
- 오탐(False Positive): 보안 도구가 취약점 또는 공격으로 판단했지만 실제 위험이 아닌 사례
- AI 에이전트: 외부 도구 호출, 코드 실행, 파일 처리, 시스템 조작 등 행동 능력을 가진 AI 시스템
- 행위 감사: 누가 또는 어떤 AI가 어떤 권한으로 어떤 조치를 수행했는지 추적·검증하는 절차

※ 출처: TTA 저널 제225호