

증거 제약 LLM 기반 보안 로그 분석 및 MITRE ATT&CK 매핑의 구성요소별 평가

 김 은 충¹ · 손 태 식^{2*}
¹아주대학교 사이버보안학과 석사과정

²아주대학교 사이버보안학과 교수

Component-wise Evaluation of Evidence-Constrained LLM-Based Security Log Analysis and MITRE ATT&CK Mapping

 Eunchong Kim¹ · Taeshik Shon^{2*}
¹Master's Course, Department of Cyber Security, Ajou University, Gyeonggi-do 16499, Korea

²Professor, Department of Cyber Security, Ajou University, Gyeonggi-do 16499, Korea

[요 약]

본 연구는 규칙 기반 보안 로그 분석이 새로운 공격 표현과 난독화에 취약한 문제와 대형 언어 모델이 근거 없는 추론이나 불안정한 출력을 생성할 수 있는 문제를 함께 다룬다. 이를 위해 입력 로그의 원문 근거를 직접 인용하는 Evidence-first LLM 분석에 Validator, 로그 유형별 Sanity-Check 및 strict evidence matching을 단계적으로 적용하고 각 구성요소가 공격 이벤트 판정과 MITRE ATT&CK Technique 매핑에 미치는 영향을 비교하였다. 실험은 합성·템플릿 기반 이벤트 4,800건과 세 종류의 LLM을 대상으로 Rule-only 및 네 가지 LLM 기반 설정을 동일한 조건에서 평가하는 방식으로 수행하였다. 실험 결과 LLM-only는 Rule-only보다 높은 이진 판별 F1을 보였고 Validator는 성능을 유지하면서 출력 구조와 ATT&CK 정합성을 확인하였다. Sanity-Check와 Full Pipeline은 오탐을 줄였으나 재현율 손실을 동반했으며 Full Pipeline의 실패 단계 중 Sanity-Check rejection이 가장 큰 비중을 차지하였다. 모델별 절대 성능에는 차이가 있었지만 hard-gate 적용에 따른 재현율 저하 경향은 공통적으로 나타났다. 이 결과는 LLM 기반 보안 로그 분석에서 단일 결합 구조의 우수성을 가정하기보다 분석 목적에 따라 검증 구성요소를 선택하고 정밀도와 재현율의 균형을 관리해야 함을 보여준다.

[Abstract]

This study addresses two related challenges: rule-based security log analysis is vulnerable to novel attack expressions and obfuscation, while large language models (LLMs) may generate unsupported reasoning or unstable outputs. We evaluate an Evidence-first LLM framework that combines a Validator, log-type-specific Sanity-Check rules, and strict evidence matching. Using 4,800 synthetic, template-based events and three LLMs, we compare a Rule-only approach with four LLM-based configurations for attack-event classification and MITRE ATT&CK Technique mapping. The LLM-only configuration achieved a higher binary-classification F1 score than Rule-only, while the Validator preserved performance and verified output structure and ATT&CK consistency. Sanity-Check and the Full Pipeline reduced false positives but lowered recall; Sanity-Check rejection accounted for the largest share of Full Pipeline failures. Although absolute performance differed across models, recall degradation under hard-gate validation was consistent. These results indicate that validation components should be selected according to analytical objectives, and that precision and recall should be balanced rather than assuming that a single architecture is universally superior.

색인어 : 보안 로그 분석, 대형 언어 모델, MITRE ATT&CK, 증거 제약 LLM, Sanity-Check

Keyword : Security Log Analysis, Large Language Model, MITRE ATT&CK, Evidence-Constrained LLM, Sanity-Check

<http://dx.doi.org/10.9728/dcs.2026.27.9.2609>


This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 11 June 2026; **Revised** 13 July 2026

Accepted 04 August 2026

***Corresponding Author, Taeshik Shon**
Tel: +82-31-219-3321

E-mail: tsshon@ajou.ac.kr

I. 서 론

엔터프라이즈 환경의 보안장비와 서비스는 네트워크 통신, 사용자 행위, 프로세스 실행 및 웹 요청과 관련된 대량의 로그를 지속적으로 생성한다. 보안관제 업무에서는 이 가운데 실제 공격과 관련된 이벤트를 신속히 선별하고, 관찰된 행위를 일관된 기준으로 설명해야 한다. 침입 탐지 시스템, SIEM 상관분석, MITRE ATT&CK 프레임워크[1], [2], Sigma 탐지 규칙[3], Suricata EVE JSON[4] 및 Splunk Common Information Model[5]은 탐지 결과를 구조화하고 분석가의 해석을 지원한다. 그러나 규칙 기반 탐지는 사전에 정의된 문자열, 정규식 및 이벤트 조건에 크게 의존한다. 새로운 공격 표현, 명령 옵션 변경, URL 또는 Base64 인코딩, 문자열 분할과 같은 난독화가 발생하면 기존 규칙이 공격 행위를 포착하지 못할 수 있다. 반대로 공격 표현과 유사한 문자열이 정상 행위에 포함되면 오탐이 발생할 수 있다. 따라서 개별 토큰의 존재 여부뿐 아니라 로그에 포함된 명령 구조와 행위 문맥을 함께 해석할 수 있는 분석 방법이 필요하다. 최근 대형 언어 모델(LLM)은 비정형 텍스트와 반정형 로그의 문맥을 해석하고, 관찰된 행위를 자연어와 구조화된 형태로 설명하는데 활용되고 있다[9]-[12]. LLM은 규칙 기반 방식이 포착하기 어려운 표현 변형과 문맥적 단서를 해석할 수 있지만, 자유 형식 출력에는 환각, 필드 누락, 근거 없는 ATT&CK Technique 제시 및 Technique-Tactic 불일치가 포함될 수 있다. 특히 공격 이벤트 판정은 후속 조사와 대응 우선순위에 직접 영향을 미치므로, LLM이 사용한 원문 근거를 추적하고 출력 구조와 ATT&CK 정합성을 검증할 필요가 있다. 보안 로그 분석 결과는 단순한 탐지 신호를 넘어 재사용 가능한 디지털콘텐츠로 활용될 수 있다. 공격 여부, ATT&CK Technique과 Tactic, 원문 evidence span 및 판단 결과를 구조화하면 SOC 대시보드의 경보 카드, 위협 인텔리전스 콘텐츠의 자동 라벨, 보안관제 리포트와 사고 대응 티켓의 입력 자료로 사용할 수 있다. 또한 공격 행위와 원문 근거를 연결한 결과는 사이버보안 교육과 훈련을 위한 사례 콘텐츠로 재구성할 수 있다. 이와 같이 본 연구는 비정형 보안 로그를 근거 추적이 가능한 정형 위협 콘텐츠로 변환한다는 점에서 디지털콘텐츠 처리·표현 기술과 연결된다. 본 연구는 LLM 기반 보안 로그 분석에서 각 검증 구성요소가 성능에 미치는 영향을 구분하여 분석한다. 이를 위해 입력 로그에서 직접 확인 가능한 문자열을 evidence로 출력하도록 요구하는 Evidence-first 프롬프트, JSON 출력 구조와 ATT&CK Technique-Tactic 관계를 확인하는 Validator, 로그 소스별 공격 단서를 검사하는 Sanity-Check 및 원문과 evidence의 정확한 일치를 확인하는 strict evidence matching을 구성하였다. 이들 요소를 Rule-only, LLM-only, LLM+ Validator, LLM+ Sanity-Check 및 Full Pipeline으로 구분하여 동일한 평가 데이터에 적용하였다.

Full Pipeline은 최종 권장 모델이 아니라 여러 검증 조건을 hard-gate로 결합했을 때의 성능 변화와 실패 지점을 확인하기 위한 strict variant로 사용하였다. 본 연구의 기여는 다음과 같다. 첫째, 입력 로그의 원문 evidence span과 공격 이벤트 판정, ATT&CK Technique, Tactic 및 confidence를 함께 생성하는 Evidence-first 구조화 출력을 정의하였다. 둘째, 출력 구조와 ATT&CK 정합성을 검사하는 Validator와 로그 소스별 공격 단서를 확인하는 Sanity-Check를 독립적인 평가 구성요소로 구분하였다. 셋째, 합성·템플릿 기반 이벤트 4,800건에 다섯 가지 설정을 적용하고 GPT-4.1-mini, Claude Haiku 4.5 및 Gemini 3.1 Flash-Lite를 이용하여 기반 모델에 따른 성능 차이를 분석하였다. 넷째, 이진 공격 이벤트 판정, ATT&CK 매핑, Technique별 성능, 난독화 조건 및 실패 단계별 오류를 함께 분석하였다. 다섯째, Full Pipeline의 실패 단계를 상호배타적으로 분류하여 재현율 저하가 주로 Sanity-Check rejection과 Technique disagreement에서 발생했음을 확인하였다. 본 논문의 구성은 다음과 같다. 2장에서는 규칙 기반 탐지, 딥러닝 기반 로그 이상 탐지 및 LLM 기반 보안 로그·ATT&CK 매핑 연구를 검토한다. 3장에서는 Evidence-first 출력, Validator, Sanity-Check 및 설정별 의사결정 규칙을 설명한다. 4장에서는 합성 평가 데이터의 생성 기준, 모델과 API 실행 환경 및 평가 지표를 기술한다. 5장에서는 구성요소별 성능, 다중 모델 비교, Technique별 결과, 난독화 조건 및 실패 단계 분석 결과를 제시하고 그 의미와 한계를 논의한다. 6장에서는 연구 결과를 요약하고 향후 연구 방향을 제시한다.

II. 관련 연구

2-1 규칙·시그니처 기반 탐지와 공통 포맷

전통적인 침입 탐지 시스템과 SIEM 상관분석은 시그니처 또는 규칙 기반으로 높은 설명 가능성을 제공한다. MITRE ATT&CK 프레임워크는 공격자의 전술과 기술을 체계적으로 분류하여 보안담당자가 탐지 규칙을 설계하고 탐지 결과를 해석하는 데 널리 활용된다[1], [2]. Sigma 프로젝트는 SIEM 환경에서 재사용 가능한 탐지규칙 템플릿을 정의하며[3], Suricata EVE JSON과 Splunk CIM은 이기종 보안장비에서 발생하는 로그를 정규화하여 상관분석과 탐지를 용이하게 한다[4], [5]. 그러나 규칙 기반 탐지는 알려진 패턴에 대한 의존성이 크고 우회 기법과 변형 공격에 취약하며 룰셋 유지관리 비용이 증가한다는 한계를 가진다. 특히 공격자가 명령 옵션을 축약하거나 URL 인코딩을 적용하거나 문자열을 분할하는 방식으로 evidence token을 변형할 경우 단순 정 규칙 기반 탐지는 탐지 범위가 크게 제한될 수 있다.

2-2 딥러닝 기반 로그 이상 탐지

규칙 기반 접근의 한계를 완화하기 위해 시스템 로그를 입력으로 하는 딥러닝 기반 이상 탐지 연구도 활발히 수행되었다. DeepLog는 LSTM 기반 시퀀스 모델을 이용해 로그 이벤트의 정상 패턴을 학습하고 정상 시퀀스로부터의 편차를 이상으로 탐지하였다[6]. LogBERT는 BERT 기반 표현 학습과 마스크 예측 오차를 활용해 로그 이상을 탐지하였다[7]. Landauer 등은 로그 이상 탐지 분야의 딥러닝 기법을 비교하고 로그 과잉 및 시퀀스 모델링의 장단점을 정리하였다[8]. 이들 연구는 정상 패턴을 학습하여 이상 행위를 자동 탐지한다는 장점이 있으나 충분한 학습 데이터가 필요하고 환경 변화에 따른 재학습 비용이 발생한다. 또한 탐지 결과를 MITRE ATT&CK Technique 수준으로 직접 해석하기 어렵다는 제약이 있다.

2-3 LLM 기반 보안 로그 분석 및 ATT&CK 매핑

최근에는 LLM을 활용하여 로그를 직접 해석하고 공격 여부, 공격 행위, ATT&CK 기법을 판별하려는 연구가 증가하고 있다[9]–[12]. 또한 MITRE ATT&CK과 이상 탐지를 결합한 공격 징후 탐지 연구는 LLM 이전 접근의 연계 맥락을 제공한다[13]. LLM은 규칙 기반 탐지보다 넓은 문맥을 활용하고 자연어 설명을 제공할 수 있으며 보안 분석가가 이해하기 쉬운 형태의 근거를 생성할 수 있다. 그러나 LLM의 자유 서술형 출력은 형식 불안정, 근거 없는 추론, ATT&CK 체계와의 불일치 문제를 포함할 수 있다. 따라서 LLM을 보안관계 환경에 적용하려면 출력 스키마, evidence grounding, 도메인 정합성 검증, 오류 억제 계층이 함께 설계되어야 한다. 최근 연구는 보안 로그 직접 분석[9]과 ATT&CK 연계 탐지[10]에서 CVE·CTI의 ATT&CK 자동 매핑과 SOC 지원으로 확장되고 있다. Crimson[14]은 CVE-to-ATT&CK 데이터와 retrieval-aware training을 제안하였고, 지식그래프 기반 CTI 구조화[15]와 TRAM:오픈소스 LLM 비교[16]는 위협 정보에서 Technique을 추출하고 구조화하는 방법을 다룬다. TTP 추출과 위협행위자 귀속[17], 규칙과 in-context learning을 결합한 TRIAGE[21]는 복수 후보와 hybrid mapping의 가능성을 평가한다. SOC 현장 연구[18]는 LLM이 고위험 결정을 대체하기보다 텔레메트리 해석과 맥락 구축을 보조하는 경향을 보고하였다. SecBench[19], R-Log[20]와 CyberSecEval 2[22]는 모델별 능력 차이, class imbalance, unseen 로그 일반화, 추론 안정성 및 안전성-효용 trade-off를 별도로 평가할 필요가 있음을 보여준다. 이러한 선행연구는 규칙·검색 결합, 구조화 출력과 ATT&CK 연계의 필요성을 제시하지만, 로그 원문 evidence, 출력 정합성 검증과 hard-gate의 단계별 영향을 동일 이벤트에서 함께 비교한 연구는 제한적이다. 본 연구는 기존 연구와 달리 LLM의 단독 성능만을 측정하지 않고 Validator와 Sanity-Check,

strict evidence matching을 단계적으로 결합하여 각 구성요소가 공격 이벤트 판정과 ATT&CK 매핑에 미치는 영향을 분리한다. 특히 Full Pipeline의 일괄적 우수성을 전제하지 않고 hard-gate가 오탐과 미탐에 미치는 영향을 구성요소별 trade-off로 해석한다. soft validation과 fallback decision은 현재 검증된 결과가 아니라 후속 연구에서 구현·평가할 설계 방향으로 한정한다.

2-4 최신 연구의 구성요소 비교

최신 LLM 기반 ATT&CK·CTI·SOC 연구를 입력 과제, 규칙·검색 결합, 구조화 출력, 원문 evidence와 검증·ablation 관점에서 비교하였다. 표 1은 체계적 문헌고찰 전체를 대체하기보다 본 연구의 구성요소와 직접 대응되는 대표 연구를 중심으로 정리한 비교표이다.

표 1. 선행연구와 본 연구의 구성 요소별 비교

Table 1. Component-wise comparison of prior studies and this study

Study	Input/Task	Pipeline/Output	Evidence/Validation Comparison
LogPrécis [9]	Unix shell attack logs	Fine-tuned LM / token-level tactic labels and attack fingerprints	Token-level annotations; model and training comparisons
Crimson [14]	CVE → ATT&CK	RAT/RAT-R, Technique	Knowledge-based validation, model/tool comparison
Fieblinger [15]	CTI → KG	Guidance/ Fine-Tuning, Triples/KG	Sentence-based evidence, method comparison
Nguyen [16]	CTI → ATT&CK	summarization + classification, Technique	TRAM-based approach, architecture comparison
TRIAGE [21]	CVE → ATT&CK	Rule + ICL, candidate Technique	CVE description, mapping/hybrid comparison
This study	Logs → attack/Technique	Rule + LLM, JSON Schema	Verbatim Evidence Span, Validator/Sanity/strict gate

본 연구의 차별점은 rule+LLM 결합 자체가 아니라 event-level verbatim evidence, ATT&CK Validator, log-type Sanity-Check와 strict gate를 동일 평가셋에서 단계별 ablation을 수행하여 오탐 억제와 재현율 손실을 분리하는 데 있다. 따라서 Full Pipeline은 최종 권장 모델이 아니라 hard-gate의 실패 지점을 확인하기 위한 strict variant로 해석한다. 보안 로그의 구조화 결과는 SOC 대시보드 정보 카드, ATT&CK 기반 위협 인텔리전스 콘텐츠, 자동화된 분석 리포트와 교육용 사례 콘텐츠로 활용될 수 있다. 이는 비

표 2. 구조화 출력 필드 명세

Table 2. Specification of structured output fields

Field	Format/Range	Evaluation Use
eventLid	string, 1–64 chars	Input/output identity validation
intrusion	integer {0,1}	Binary detection
technique	1–2 IDs, T1059/T1190/T1047/T1071/T0000	Top–1 macro–F1 and supplementary candidate hit
tactic	1–2 IDs, TA0001/TA0002/TA0011/TA0000	Technique–Tactic consistency validation
evidence	0–4 strings	Exact–substring strict evidence–gate evaluation
confidence	number, 0–1	Stored as an uncalibrated auxiliary signal only

표 3. Validator 규칙 17개와 strict evidence gate 1개

Table 3. Seventeen validator rules and one strict evidence gate

Validation Rule	Action	Strict Evidence Gate
not object	invalid → negative	No
missing required field	invalid → negative	No
eventLid mismatch	invalid → negative	No
invalid intrusion type or value	invalid → negative	No
invalid technique container	invalid → negative	No
invalid tactic container	invalid → negative	No
invalid evidence container	invalid → negative	No
Technique–Tactic Length Mismatch	invalid → negative	No
invalid confidence	invalid → negative	No
invalid technique id format	invalid → negative	No
unknown technique	invalid → negative	No
deprecated or revoked technique	invalid → negative	No
unknown tactic	invalid → negative	No
Technique–Tactic Mismatch	invalid → negative	No
normal label in attack output	invalid → negative	No
attack label in normal output	invalid → negative	No
missing evidence for attack	invalid → negative	No
exact evidence substring match	Full gate	Yes

정형 보안 데이터를 근거 추적이 가능한 정형 디지털콘텐츠로 변환한다는 점에서 디지털콘텐츠 처리·표현 기술과도 연결된다.

III. 증거 제약 LLM 분석 파이프라인 및 구성요소

3-1 파이프라인 설계

본 파이프라인은 JSONL 기반 로그 정규화, Evidence-first 프롬프트, LLM 구조화 출력, Validator, Sanity-Check, 최종 의사결정 단계로 구성된다. 핵심 목적은 LLM의 문맥 해석 능력을 활용하되, 보안 관제에서 요구되는 출력 일관성, 근거 추적성, ATT&CK 정합성, 오답 억제 기능을 구조적으로 보완하는 것이다. JSONL 기반 정규화 단계에서는 이기종 보안 로그를 이벤트 단위 JSON 객체로 변환한다. 각 이벤트는 로그 소스, 명령행 또는 URI, 프로세스명, 네트워크 메타데이터, 원문 메시지 등 분석에 필요한 공개 필드를 포함한다. 정답 라벨, 난독화 수준, template split과 후보 Technique은 API 입력에서 분리하여 label leakage를 방지하였다. Evidence-first 프롬프트는 LLM이 공격 여부와 ATT&CK Technique을 판단할 때 입력 로그에서 직접 관찰 가능한 evidence를 함께 출력하도록 요구한다. 출력은 event_id, intrusion, technique, tactic, evidence, confidence 필드를 포함하는 JSON 형식으로 제한한다. confidence는 자기 보고형 보조 신호이며 확률값으로 간주하지 않는다. Validator는 JSON 파싱, 필수 필드, 필드 타입, event identity, technique ID와 Technique–Tactic 정합성을 검사한다. Sanity-Check는 로그 유형별 evidence rule을 기반으로 LLM의 공격 판단을 보수적으로 보정한다. 본 연구에서는 Rule-only, LLM-only, LLM+ Validator, LLM+ Sanity-Check 및 Full Pipeline의 다섯 설정을 비교한다. Full Pipeline은 Validator, Sanity-Check, rule Technique 일치와 strict evidence matching을 순차적으로 요구하는 가장 엄격한 설정이며 최종 권장 모델이 아니라 hard-gate의 효과와 실패 지점을 분석하기 위한 strict variant로 정의한다.

3-2 Evidence-first 프롬프트와 구조화 출력 스키마

재현 패키지의 실제 system/user prompt를 사용하였다. public event에는 event_id, log_type, timestamp, host, user, src_ip, dst_ip, process, command, uri, protocol과 message만 포함하고, 정답·난독화·template split·candidate label은 private label 파일로 분리하였다. runner는 중첩 필드를 포함해 private field가 발견되면 호출을 중단한다.

[System prompt 핵심]

Use only information directly observable in the supplied normalized event. Allowed labels are T1059-TA0002, T1190-TA0001, T1047-TA0002, T1071-TA0011, and benign T0000-TA0000. intrusion=1 requires direct evidence. Return one primary Technique and optionally one secondary candidate. Every evidence item must be a non-empty character-for-character substring copied from message. Return only the structured object.

[User prompt]

Analyze the following normalized security event. Copy every evidence span exactly from the message field.

{{EVENT_JSON}}

3-3 Validator와 Sanity-Check 규칙

표 2는 LLM 구조화 출력에 포함되는 여섯 개 필드의 형식, 허용 범위 및 평가 용도를 제시한다. event_id와 intrusion은 각각 입력-출력 동일성 검증과 이진 공격 이벤트 판정에 사용하고 technique과 tactic은 ATT&CK 매핑 및 정합성 검증에 활용한다. evidence는 strict evidence gate의 원문 부분 문자열 검사에 사용하며 confidence는 보정된 확률이 아닌 보조 신호로만 저장한다. 표 3은 이러한 출력을 검사하는 17개 Validator 규칙과 Full Pipeline에서만 적용하는 strict evidence gate를 구분한다. T0000은 정상 이벤트용 dummy label이며 sub-technique, deprecated 또는 revoked gold label은 사용하지 않았다.

표 4는 WAF 4개, EDR 5개 및 Proxy 3개로 구성된 총 12개의 source-specific Sanity-Check 규칙을 제시한다. 각 규칙은 로그 소스에서 직접 관찰 가능한 문자열이나 행위 패턴을 ATT&CK Technique과 연결한다. 이 규칙 집합은 전체 ATT&CK를 포괄하는 운영 룰셋이 아니라 구성요소 비교를 위한 제한된 평가 규칙이다. 하나 이상의 규칙이 일치하면 rule match로 판정하고 일치한 Technique 집합을 유지한다. Full Pipeline은 LLM Top-1이 이 집합에 포함되는지를 확인하며 Rule-only에서 단일 대표 Technique이 필요한 경우에는 표 4에 제시된 규칙 순서에 따라 최초로 일치한 규칙의 Technique을 사용한다. 공격 rule coverage는 71.50%, 정상 rule match 비율은 11.75%였다.

3-4 구성별 의사결정 규칙

① Rule-only: source-specific rule 결과를 그대로 사용한다. ② LLM-only: 구조화 출력의 Top-1 intrusion/Technique을 사용한다. ③ LLM+ Validator: hard-error output만 negative로 전환한다. ④ LLM+ Sanity-Check:

표 4. 로그 유형별 Sanity-Check 전체 rule set(12개)

Table 4. Complete log-type-specific sanity-check rule set (12 rules)

Rule ID	Source	Tech.	Representative Meaning
WAF-T1190-SQLI-01	WAF	T1190	SQL injection UNION SELECT pattern
WAF-T1190-TRAVERSAL-01	WAF	T1190	Path traversal or sensitive file request
WAF-T1190-TEMPLATE-01	WAF	T1190	Public-facing application exploitation marker
WAF-T1059-CMDINJ-01	WAF	T1059	Command interpreter token in an HTTP parameter
EDR-T1059-POWERSHELL-01	EDR	T1059	PowerShell interpreter with execution or encoded-command indicators
EDR-T1059-CMD-01	EDR	T1059	Windows command shell invocation
EDR-T1047-WMIC-01	EDR	T1047	WMIC Win32_Process creation
EDR-T1047-CIM-01	EDR	T1047	PowerShell WMI/ CIM method invocation
EDR-T1071-WEB-01	EDR	T1071	Process command line contacting a web-protocol endpoint
PROXY-T1071-CALLBACK-01	Proxy	T1071	Periodic web callback path
PROXY-T1071-PERIODIC-01	Proxy	T1071	Long-lived or periodic application-layer traffic
PROXY-T1190-EXPLOIT-01	Proxy	T1190	Exploit marker observed at a reverse proxy

LLM intrusion=1과 Sanity rule match를 모두 요구한다.

⑤ Full Pipeline: Validator valid, Sanity match, rule Technique와 LLM Top-1 일치, 모든 evidence exact substring을 순차적으로 요구한다. strict evidence matching은 Sanity-Check와 별개이며 LLM evidence가 message의 비어 있지 않은 정확한 substring인지 검사한다. 이 조건은 Full Pipeline에서만 hard-gate로 적용한다. 누락·refusal·incomplete·parse error·invalid output은 사전 정의에 따라 negative(T0000)로 평가한다. 모델 응답은 이벤트별 첫 terminal response를 고정하고 transport/rate-limit/server 오류만 최대 5회 재시도한다. Full Pipeline은 제안 구조의 최종 권장형이 아니라 hard-gate의 효과와 실패 지점을 확인하는 strict variant이다. soft validation, weighted scoring과 fallback decision은 본 연구에서 구현·평가하지 않았다.

그림 1은 Validator와 Sanity-Check가 하나의 고정 직렬 구조가 아니라 실험 설정별 비교 경로임을 구분한다. Full Pipeline에서는 Validator와 Sanity-Check 이후 rule Technique와 LLM Top-1의 일치 및 strict evidence matching을 hard AND 조건으로 적용한다. 점선의 weighted soft validation과 fallback/분석가 검토 경로는 본 연구에서 구현·평가하지 않은 후속 연구 방향이다.

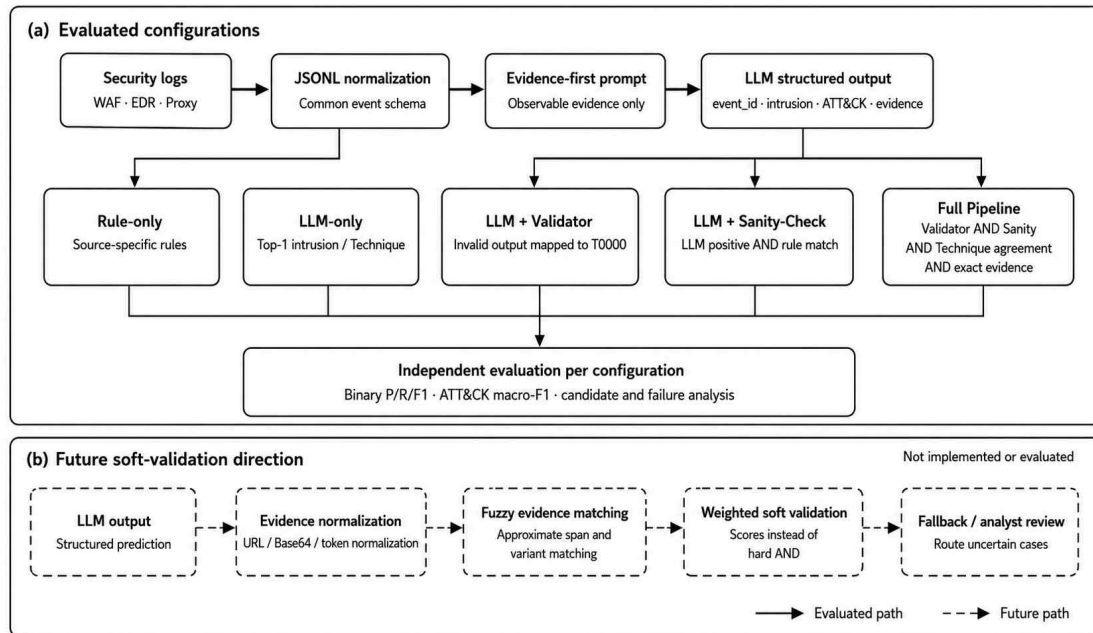


그림 1. 제안 파이프라인 구조와 실험 설정별 경로 및 향후 연구 방향

Fig. 1. Proposed pipeline architecture, configuration-specific experimental paths, and future research directions

3-5 공격 이벤트 판정 및 ATT&CK 기법 매핑 평가 정의

최종 의사결정은 설정별로 다르게 수행된다. LLM-only에서는 LLM이 출력한 intrusion 값을 그대로 사용한다. LLM+ Validator에서는 Validator를 통과한 LLM 출력을 사용한다. LLM+ Sanity-Check에서는 LLM의 intrusion 판단과 Sanity-Check의 evidence rule 결과를 결합하여 고정밀 판단을 수행한다. Full Pipeline에서는 Validator 통과, Sanity-Check 일치, rule Technique와 LLM Top-1 일치, strict evidence matching을 모두 만족하는 경우에만 공격 이벤트로 판정한다. 이 hard AND 결합은 오탐 억제에는 유리하지만 하나의 조건이라도 실패하면 실제 공격 이벤트가 false negative로 전환될 수 있다. 평가 지표는 이진 공격 이벤트 판정과 ATT&CK 기법 매핑으로 구분한다. 이진 공격 이벤트 판정은 TP, FP, TN, FN을 기반으로 Precision, Recall, F1-score를 계산한다. ATT&CK 기법 매핑은 Technique ID를 클래스 라벨로 보고 macro F1을 계산한다. T0000은 클래스 평균에서 제외한다. Ambiguous 이벤트의 primary label 외 후보 기법은 candidate-aware 보조 지표로 별도 보고한다.

IV. 실험 설정

4-1 데이터셋 구성

본 연구의 평가는 총 4,800개 이벤트를 대상으로 수행하였

다. 이 중 공격 이벤트는 3,600개, 정상 이벤트는 1,200개로 구성된다. 평가는 Rule-only, LLM-only, LLM+ Validator, LLM+ Sanity-Check, Full Pipeline의 다섯 가지 설정을 동일한 이벤트 집합에 적용한 뒤, 이진 공격 이벤트 판정 성능과 ATT&CK 기법 매핑 성능을 비교하는 방식으로 수행하였다. 평가 데이터는 seen template, unseen template, benign near-miss, hard obfuscation, ambiguous candidate 등 다양한 조건을 포함한다. seen template은 비교적 명확한 공격 표현을 포함하는 조건이며 unseen template은 사전에 정의된 규칙 또는 예시와 다른 변형을 포함한다. benign near-miss는 공격과 유사한 토큰을 포함하지만 실제로는 정상 이벤트로, 오탐 억제 능력을 평가하는 데 사용된다. hard obfuscation은 문자열 분할, 인코딩, 경로 교란 등 강한 난독화가 적용된 조건이다. ambiguous candidate는 하나의 이벤트가 복수의 ATT&CK Technique 후보와 연결될 수 있는 조건이다. 난독화 강도는 None, Easy, Medium, Hard의 네 단계로 구분하였다. None은 원문 명령 또는 페이로드를 그대로 사용한 조건이고 Easy는 옵션 축약, 대소문자 변화, 단순 치환을 포함한다. Medium은 부분 인코딩, 공백 삽입, 주석 삽입과 같은 중간 수준 변형을 포함한다. Hard는 URL/Base64 인코딩, 문자열 분할, 경로 분할, 명령 토큰 교란 등 강한 변형을 포함한다.

4-2 비교 설정별 판정 절차

모든 모델과 비교 설정에는 동일한 이벤트 ID, 데이터셋 해시, 프롬프트, ATT&CK 매핑, Validator, Sanity-Check 및

표 5. 로그 소스별 분포

Table 5. Distribution of events by log source

Source	Subtype	Attack	Near-miss	Routine	Total
WAF	WAF_HTTP	900	152	148	1200
EDR	EDR_PROCESS EDR_PROCESS_NETWORK	1800	307	293	2400
Proxy	PROXY_HTTP PROXY_REVERSE_HTTP	900	141	159	1200

평가 스크립트를 적용하였다. LLM-only는 구조화 출력의 Top-1 intrusion/Technique을 사용하고, LLM+ Validator는 hard-error 출력만 negative로 변환한다. LLM+ Sanity-Check는 LLM의 intrusion 값이 1이고 Sanity-Check 규칙이 일치하는 경우에만 공격 이벤트로 판정한다. Full Pipeline은 Validator 통과, Sanity-Check 규칙 일치, 규칙 Technique과 LLM Top-1 Technique의 일치, 모든 evidence의 원문 내 정확한 부분 문자열 일치를 순차적으로 요구한다. 응답 누락, 거부, 미완료 또는 유효하지 않은 출력은 사전에 정한 기준에 따라 정상(T0000)으로 처리한다. 각 이벤트의 첫 최종 응답을 평가에 사용하였으며, 전송 오류, 요청 한도 초과 및 서버 오류에 대해서만 최대 5회 재시도하였다.

4-3 데이터셋 출처·구성 및 조건 생성 기준

표 5는 WAF, EDR 및 Proxy 로그 소스별로 공격, benign near-miss 및 routine 이벤트의 분포를 제시한다. EDR이 2,400건으로 전체의 절반을 차지하고 WAF와 Proxy는 각각 1,200건으로 구성된다. 전체 4,800개 이벤트는 공격 3,600건과 정상 1,200건의 결정적 synthetic/template 기반 데이터이며 실제 SOC 로그나 공개 원시 데이터셋을 직접 사용하지 않았다. public input과 private label은 물리적으로 분리하고 label leakage guard를 적용하였다. 따라서 이 분포는 세 로그 소스에서 공격과 정상 유사 표현을 비교하기 위한 통제 조건이며 실제 SOC의 장비별 발생 비율을 대표하지 않는다.

4-4 라벨 분포 및 ATT&CK 평가 기준

표 6은 seen, unseen, hard obfuscation, ambiguous candidate 및 두 정상 조건의 모집단, 이벤트 수와 생성 기준을 정리한다. Seen과 unseen은 rule design의 held-in/held-out 표현군을 구분하며 LLM pretraining 관측 여부를 의미하지 않는다. Hard 조건은 인코딩, 명령 토큰 분할과 경로 교란을 적용하고, ambiguous 조건은 동일 이벤트에 secondary ATT&CK behavior의 직접 근거를 포함한다. Hard 710건과 ambiguous 360건은 template split과 중첩될 수 있으므로 표의 건수를 단순 합산하여 전체 이벤트 수로 해석해서는 안 된다. 예시는 seen의 WMIC process

표 6. 조건별 생성 기준과 건수

Table 6. Generation criteria and event counts by condition

Condition	Population	Count	Generation/Construction Criteria
seen_template	attack	2,520	Uses held-in expression families represented in the source-specific rule design and canonical examples.
unseen_template	attack	1,080	Uses separate held-out expression families not used as canonical rule examples; labels and source distributions are unchanged.
hard_obfuscation	attack	710	Applies URL/double encoding, command-token splitting, caret/backtick insertion, Base64 encoding, and path/token perturbation.
ambiguous_candidate	attack	360	Places direct evidence of a secondary ATT&CK behavior in the same event; the deterministic primary label remains the main score.
benign_near_miss	benign	600	Contains attack-like lexical tokens in a non-adversarial operational, documentation or administrative context.
benign_routine	benign	600	Contains ordinary web, process, or proxy activity without a directly represented attack behavior.

표 7. Technique-Tactic 라벨 분포(ATT&CK v19.1)

Table 7. Technique-tactic label distribution (ATT&CK v19.1)

Tech	Technique name	Tactic	Support	WAF	EDR	Proxy
T1059	Command and Scripting Interpreter	TA0002	900	300	600	0
T1190	Exploit Public-Facing Application	TA0001	900	600	0	300
T1047	Windows Management Instrumentation	TA0002	900	0	900	0
T1071	Application Layer Protocol	TA0011	900	0	300	600

create, unseen의 Win32_Process Call Create, hard의 이중 인코딩 /beacon 경로, ambiguous의 /beacon과 UNION SELECT 동시 포함이다.

4-5 평가 기준

표 7은 네 Technique의 Tactic, 클래스별 support와 로그 소스 분포를 제시한다. 각 클래스는 900건으로 균형화되었지만 적용 로그 소스는 서로 다르다. 표 8은 로그 소스, Technique 및 난독화 수준의 교차 분포를 보여주며 난독화 별 성능을 해석할 때 표본 구성의 영향을 확인하는 기준으로 사용한다. 평가 기준은 다음과 같다. ① 공격 클래스 수는 K=4이며 각 클래스 support는 900건이다. ② T0000은 정

표 8. 로그 소스·Technique·난독화 교차 분포

Table 8. Cross-distribution of log sources, techniques, and obfuscation levels

Condition	WAF	EDR	Proxy	T1059	T1190	T1047	T1071	Attack	Benign
None	254	533	246	259	258	258	258	1033	0
Easy	310	597	302	302	303	302	302	1209	0
Medium	159	318	171	162	162	162	162	648	0
Hard	177	352	181	177	177	178	178	710	0
benign_near_miss	152	307	141	0	0	0	0	0	600
benign_routine	148	293	159	0	0	0	0	0	600

상 dummy label로 macro 평균에서 제외한다. ③ 주 ATT&CK 지표는 전체 4,800건의 end-to-end macro-F1 이고 공격 3,600건의 mapping macro-F1을 보조 보고한다. ④ support<20은 rare class로 표시하되 제외하지 않고 zero division은 0으로 처리한다. ⑤ sub-technique, deprecated 또는 revoked gold label은 없다. ⑥ ambiguous 이벤트는 deterministic primary label exact Top-1을 주 점수로 사용하고 Candidate Hit@1/2를 보조 지표로 보고한다. ⑦ 난독화별 N은 공격 이벤트 수이며 정상 이벤트는 해당 Recall 계산에서 제외한다. 전체 이벤트 기반 ATT&CK macro-F1은 정상 이벤트에서의 공격 Technique 예측도 각 클래스의 FP에 포함한다. Paired statistical test와 bootstrap confidence interval은 수행하지 않았으므로 소폭 차이를 통계적 유의성으로 해석하지 않는다. 데이터 생성 결과 prompt, mapping, 규칙, 평가 스크립트와 SHA-256 manifest는 재현 패키지에 포함하였다.

4-6 모델 및 API 실행 설정

표 9는 GPT-4.1-mini, Claude Haiku 4.5 및 Gemini 3.1 Flash-Lite의 모델명, 구조화 출력 방식, decoding 설정과 실행일을 제시한다. 세 모델은 동일한 4,800개 event ID, dataset hash, prompt hash, ATT&CK mapping, Validator, Sanity-Check 및 평가 스크립트를 사용하였다. 공급자별 API 문법과 Structured Output 방식은 다르지만 모든 응답을 동일 내부 스키마로 변환해 같은 평가 코드로 처리하였으므로 모델별 결과 차이는 평가 절차가 아니라 기반 모델과 공급자별 응답 특성의 차이로 해석할 수 있다. 공통 max output tokens=512, timeout=120초,

표 9. 모델 및 API 실행 설정

Table 9. Model and API execution settings

Provider/Model	Structured Output	Temp./top_p	Execution Date
GPT-4.1-mini	Responses/Pydantic	0.1/1.0	2026-07-17
Claude Haiku 4.5	Messages/Pydantic	0.1/Omitted	2026-07-17
Gemini 3.1 Flash-Lite	JSON Schema	0.1/1.0	2026-07-17

concurrency=2였고 누락, refusal, incomplete 및 invalid output은 보수적 negative로 평가하였다.

V. 실험 결과 및 분석

5-1 GPT-4.1-mini 구성별 전체 성능

표 10. GPT-4.1-mini 기반 구성별 전체 성능 비교

Table 10. Overall performance comparison across GPT-4.1-mini-based configurations

Method	P	R	Bin F1	ATT&CK E2E F1	Attack-Only F1
Rule-only	0.9481	0.7150	0.8152	0.6934	0.7048
LLM-only	0.9013	0.8450	0.8723	0.6851	0.7193
LLM+Validator	0.9013	0.8450	0.8723	0.6851	0.7193
LLM+Sanity-Check	0.9527	0.5711	0.7141	0.5459	0.5543
Full Pipeline	0.9426	0.4608	0.6190	0.5153	0.5243

표 11. GPT-4.1-mini 기반 구성별 혼동행렬 비교

Table 11. Confusion matrix comparison across GPT-4.1-mini-based configurations

Method	TP	FP	TN	FN
Rule-only	2,574	141	1,059	1,026
LLM-only	3,042	333	867	558
LLM+Validator	3,042	333	867	558
LLM+Sanity-Check	2,056	102	1,098	1,544
Full Pipeline	1,659	101	1,099	1,941

5-2 기반 모델 비교와 반복 안정성

표 12는 동일한 4,800개 이벤트에 세 LLM을 적용하고 모델별 LLM-only, LLM+Sanity-Check 및 Full Pipeline 성능을 비교한다. Claude와 Gemini의 LLM-only Binary F1은 GPT보다 높았지만 세 모델 모두 Sanity-Check와 Full Pipeline을 적용할수록 Recall과 Binary F1이 감소하였다. 이는 절대 성능은 기반 모델에 의존하지만 hard-gate 적용에

표 12. 동일한 4,800개 이벤트에 대한 모델 및 구성 비교

Table 12. Cross-model and configuration comparison on the same 4,800 events

Model	Method	P	R	F1	ATT&CK E2E F1	Validation Rate	API Success Rate
GPT-4.1-mini	LLM-only	0.9013	0.8450	0.8723	0.6851	1.0000	1.0000
GPT-4.1-mini	LLM+Sanity-Check	0.9527	0.5711	0.7141	0.5459	1.0000	1.0000
GPT-4.1-mini	Full Pipeline	0.9426	0.4608	0.6190	0.5153	1.0000	1.0000
Claude Haiku 4.5	LLM-only	0.9151	0.8981	0.9065	0.7799	1.0000	1.0000
Claude Haiku 4.5	LLM+Sanity-Check	0.9640	0.6242	0.7577	0.6307	1.0000	1.0000
Claude Haiku 4.5	Full Pipeline	0.9559	0.5053	0.6611	0.6039	1.0000	1.0000
Gemini 3.1 Flash-Lite	LLM-only	0.8763	0.9619	0.9171	0.7792	0.9998	0.9998
Gemini 3.1 Flash-Lite	LLM+Sanity-Check	0.9523	0.6769	0.7914	0.6423	0.9998	0.9998
Gemini 3.1 Flash-Lite	Full Pipeline	0.9439	0.5700	0.7108	0.6185	0.9998	0.9998

표 13. GPT-4.1-mini의 480개 이벤트 부분집합 3회 반복 결과

Table 13. Three repeated runs of GPT-4.1-mini on a 480-event subset

Run	P	R	Bin F1	ATT&CK E2E F1	Attack-Only F1
Run 1	0.9112	0.8556	0.8825	0.6997	0.7305
Run 2	0.9139	0.8556	0.8838	0.6926	0.7226
Run 3	0.9110	0.8528	0.8809	0.7013	0.7318

다른 Recall 저하 방향은 공통적임을 보여준다. Validation Rate와 API Success Rate가 거의 1.0이므로 주요 차이는 구조화 출력 실패보다 모델의 분류·매핑 특성에서 발생한 것으로 해석한다.

표 13은 로그 소스, 이진 라벨, Technique 및 난독화 조건의 분포를 유지하도록 고정 seed로 층화 추출한 480개 이벤트를 GPT-4.1-mini로 세 차례 반복 실행한 결과를 제시한다. Binary F1은 0.8809-0.8838, 평균±표준편차는 0.8824 ± 0.0015 였고 ATT&CK end-to-end macro-F1은 0.6979 ± 0.0046 이었다. Joint prediction agreement는 동일 이벤트에서 intrusion과 Top-1 Technique이 모두 일치한 비율로 정의했으며 0.9625-0.9667을 기록하였다. 이는 해당 부분집합에서 비교적 안정적인 출력 경향을 보였지만 전체 데이터나 다른 모델의 안정성으로 일반화할 수는 없다. GPT와 Claude는 4,800/4,800 success, Gemini는 4,799 success와 1 incomplete였으며 incomplete 1건은 사전 정책에 따라 negative로 평가하였다.

5-3 전체 성능 해석

표 10은 GPT-4.1-mini를 기준으로 Rule-only부터 Full Pipeline까지 다섯 구성의 이진 공격 이벤트 판정과 ATT&CK 매핑 성능을 비교한다. LLM-only는 Rule-only보다 Binary Recall과 F1을 높였고 Validator 적용 후에도 동일한 성능을 유지하였다. 반면 LLM+Sanity-Check와 Full Pipeline은

Precision을 높이는 대신 Recall을 낮춰 구조적 제약의 강화가 오탐과 미탐 사이의 trade-off를 유발함을 보여준다. 표 11은 이러한 차이를 TP, FP, TN 및 FN으로 분해한다. LLM-only는 TP를 2,574건에서 3,042건으로 늘렸지만 FP도 141건에서 333건으로 증가하였다. LLM+Sanity-Check는 FP를 102건으로 줄였으나 FN이 1,544건으로 증가했고, Full Pipeline은 FN 1,941건을 기록해 가장 엄격한 설정에서 Recall 손실이 커졌음을 확인할 수 있다. ATT&CK end-to-end macro-F1과 attack-only mapping macro-F1의 방향이 달라 이진 판정과 Technique 매핑을 별도로 해석해야 한다.

5-4 난독화 강도별 성능 분석

표 14는 난독화 수준별로 Rule-only, LLM-only, LLM+Sanity-Check 및 Full Pipeline의 이진 Recall과 공격 이벤트 기반 ATT&CK mapping macro-F1을 비교한다. GPT LLM-only의 이진 Recall은 None 0.8451, Easy 0.8478, Medium 0.8349, Hard 0.8493으로 유사했지만 mapping macro-F1은 None 0.7660에서 Hard 0.5445로 감소하였다. 따라서 Hard 조건의 높은 이진 Recall만으로 Technique 매핑까지 강건하다고 해석할 수 없다. Rule-only의 Hard Recall/F1은 0.6930/0.3802, Full Pipeline은 0.3479/0.1688이었다. 표의 N은 해당 난독화 수준의 공격 이벤트 수이며 정상 1,200건은 계산에서 제외하였다.

표 14. 난독화 수준별 Recall 및 공격 이벤트 기반 ATT&CK Macro-F1

Table 14. Recall and attack-only ATT&CK Macro-F1 by obfuscation level

Level	N	Rule R/F1	LLM R/F1	S-C R/F1	Full R/F1
None	1033	0.7367/ 0.7892	0.8451/ 0.7660	0.5915/ 0.6066	0.4985/ 0.6082
Easy	1209	0.7361/ 0.7892	0.8478/ 0.7528	0.6022/ 0.6043	0.5161/ 0.6036
Medium	648	0.6651/ 0.6913	0.8349/ 0.7549	0.5123/ 0.5839	0.4213/ 0.5436
Hard	710	0.6930/ 0.3802	0.8493/ 0.5445	0.5423/ 0.3509	0.3479/ 0.1688

5-5 Technique별·조건별 성능

표 15는 GPT-4.1-mini 주 실험에서 네 Technique의 support와 Rule-only 및 LLM-only의 클래스별 성능을 제시한다. LLM-only는 T1059, T1190 및 T1047에서 Rule-only보다 높은 F1을 기록했지만 T1071의 Recall은 0.1611, F1은 0.2775로 크게 낮았다. 이는 LLM-only의 end-to-end ATT&CK macro-F1이 소폭 낮아진 주요 원인이다. 표 16은 seen, unseen, hard obfuscation 및 ambiguous 조건별로 각 구성의 이진 Recall과 공격 이벤트 기반 mapping F1을 비교한다. LLM-only는 네 조건에서 높은 Recall을 유지했지만 Full Pipeline은 unseen과 hard obfuscation 조건에서 Recall과 mapping F1이 크게 낮아져 hard-gate의 취약성이 template coverage와 난독화 조건에서 두드러짐을 보여준다.

표 15. GPT-4.1-mini 주 실험의 Technique별 Support, Rule-only F1 및 LLM-only Precision/Recall/F1

Table 15. Per-technique support, rule-only F1, and LLM-only precision/recall/F1 in the primary GPT-4.1-mini experiment

Tech.	Support	Rule F1	LLM P/R/F1
T1059	900	0.5310	0.5964/0.8933/0.7153
T1190	900	0.5161	0.8387/0.9478/0.8899
T1047	900	0.8166	0.8751/0.8411/0.8578
T1071	900	0.9098	1.0000/0.1611/0.2775

표 16. 공격 조건 및 구성별 Recall과 공격 이벤트 기반 Mapping F1

Table 16. Recall and attack-only mapping F1 by attack condition and configuration

Condition	N	Rule R	LLM R/MapF1	S-C R	Full R/MapF1
Seen	2520	0.8111	0.8587/ 0.7527	0.6718	0.5730/ 0.6441
Unseen	1080	0.4907	0.8130/ 0.6375	0.3361	0.1991/ 0.1538
Hard Obf.	710	0.6930	0.8493/ 0.5445	0.5423	0.3479/ 0.1688
Ambiguous	360	0.7417	0.9917/ 0.6413	0.7417	0.4417/ 0.4967

표 17. Ambiguous candidate 이벤트 보조 평가(360건)

Table 17. Supplementary evaluation of ambiguous candidate events (360 Events)

Method	Primary Top-1	Candidate Hit@1/2
Rule-only	0.5917	0.7417
LLM-only	0.6917	0.9917
LLM+Validator	0.6917	0.9917
LLM+Sanity-Check	0.4889	0.7417
Full Pipeline	0.3861	0.4417

표 17은 360개의 ambiguous candidate 이벤트에서 deterministic primary label 기준 Top-1 정확도와 후보 집합을 고려한 Candidate Hit@1/2를 비교한다. GPT LLM-only의 primary Top-1은 0.6917이지만 Candidate Hit@1/2는 0.9917이었다. 이는 모델이 이벤트에 직접 표현된 유효 후보 Technique을 선택했더라도 사전에 지정한 primary label과 순위가 다르면 exact Top-1 평가에서 오답으로 처리될 수 있음을 보여준다. 주 ATT&CK macro-F1은 primary exact Top-1을 유지하고 candidate-aware 결과는 보조 지표로만 보고한다.

5-6 Full Pipeline 오류 유형 분석

표 18은 GPT-4.1-mini Full Pipeline에서 발생한 FN 1,941건을 최초 실패 단계에 따라 상호배타적으로 분류한 결과를 나타낸다. Sanity-Check rejection이 986건 (50.80%)으로 가장 큰 비중을 차지했고, LLM intrusion miss/refusal 558건과 Technique disagreement 373건이 뒤를 이었다. Strict evidence mismatch는 24건에 불과하여 Full Pipeline의 주요 병목이 exact evidence matching 자체보다 source-specific rule coverage와 LLM-rule Technique agreement에 있음을 보여준다. 자동 분류 우선순위는 LLM miss/refusal, Validator rejection, Sanity-Check rejection, Technique disagreement, strict evidence mismatch, Other 순이며

표 18. GPT-4.1-mini Full Pipeline의 상호 배타적 실패 단계

Table 18. Mutually exclusive failure stages of the GPT-4.1-mini full pipeline

Failure Stage	Count	FN Ratio	Decision Rule
LLM intrusion miss/refusal	558	28.75%	LLM intrusion ≠ 1 or unusable output
Validator rejection	0	0.00%	Schema, ID, or tactic hard error
Sanity-Check rejection	986	50.80%	Source-specific rule mismatch
Technique disagreement	373	19.22%	LLM Top-1 not in matched rule techniques
Strict evidence mismatch	24	1.24%	Evidence is not an exact substring

Note: Percentages are rounded to two decimal places and may not sum to 100%.

각 FN은 최초로 충족한 하나의 단계에만 배정한다. 다섯 유형의 합은 혼동행렬의 FN 1,941건과 일치하고 Other final-decision FN은 0건이었다.

표 19는 표 18의 오류 유형을 판정하는 우선순위와 상호 배타성 기준을 정의한다. 각 FN은 앞선 단계에 해당하지 않을 때만 다음 단계로 이동하며 최초로 충족한 실패 유형 하나에만 배정된다. 이 자동 규칙은 한 이벤트가 여러 오류 유형에 중복 집계되는 것을 방지한다. 조건별 FN(seen, unseen, hard, ambiguous)은 별도의 비배타적 보조 집계이므로 실패 단계 표와 혼합하지 않는다.

표 19. 오류 분류 우선순위와 배타성

Table 19. Priority and mutual exclusivity of error classification

Priority	Failure Stage	Automated Rule	Exclusive
1	LLM intrusion miss/refusal	truth_intrusion=1 AND Full_Pipeline=0 AND llm_intrusion!=1	Yes
2	Validator rejection	Prior stage false AND validator_valid=false	Yes
3	Sanity-Check rejection	Prior stages false AND sanity_matched=false	Yes
4	Technique disagreement	Prior stages false AND LLM Top-1 Technique not in matched rule techniques	Yes
5	Strict evidence mismatch	Prior stages false AND not all evidence spans are exact non-empty substrings of message	Yes
6	Other final-decision FN	All prior stages false but Full Pipeline remains negative	Yes

5-7 논의 및 한계

첫째, GPT LLM-only는 Rule-only보다 Recall과 Binary F1을 높였지만 Precision을 낮추고 FP를 증가시켰다. ATT&CK 성능도 end-to-end와 attack-only mapping에서 방향이 달랐다. 따라서 “LLM-only가 모든 지표에서 우수하다”가 아니라 탐지 누락 감소와 정상 near-miss 오탐 사이의 trade-off로 해석해야 한다. 둘째, Validator는 schema-constrained API 환경에서 GPT·Claude 4,800건과 Gemini 4,799건을 valid로 판정하여 성능을 거의 바꾸지 않았다. 그 가치는 F1 상승보다 identity·type·허용 ID·Technique·Tactic 정합성과 후속 자동화 안전성에 있다. 셋째, Sanity-Check는 세 모델 모두 Precision을 높이고 FP를 줄였으나 Recall을 낮췄다. Full Pipeline의 주 병목도 Sanity-Check rejection과 Technique disagreement였다. 앞선 검증 단계를 통과한 뒤 strict evidence mismatch로 탈락한 FN은 24건이었다. 따라서 다른 판정 조건을 유지한 채 마지막 evidence 검사만 완화하는 것으로는 전체 FN 감소에 한계가 있으며, rule coverage와 multi-Technique agreement도 함께 검토해야 한다. 넷째, 데이터는 결정적

synthetic/template 기반이고 실제 SOC 로그를 포함하지 않는다. 공격:정상 3:1, 네 Technique×각 900건의 균형 구조는 실제 base rate, class imbalance, 장비별 누락과 시간 상관을 반영하지 못한다. seen/unseen도 LLM pretraining의 관측 여부가 아니라 rule design의 held-in/held-out 표현군이다. 다섯째, soft validation, fuzzy matching, weighted scoring과 fallback decision은 구현·평가하지 않았다. 따라서 오류 분석에 근거한 후속 연구 가설로만 제시한다. confidence도 보정 확률이 아니다. 재현 패키지에는 public event와 private label 분리본, 실제 system/user prompt, JSON schema, ATT&CK Enterprise v19.1 mapping, Validator 17개 규칙, strict evidence gate 1개, Sanity-Check 12개 규칙, 모델별 실행 manifest, 평가 및 표 생성 스크립트, 결과 CSV와 SHA-256 manifest를 포함하였다. API key, 결제 정보 및 공급자 계정 정보는 포함하지 않았다. 재현 데이터와 실행·평가 코드는 GitHub 공개 저장소(<https://github.com/ojoito-cell/Evidence-Constrained-LLM-Security-Log-Re-experiment>)를 통해 제공한다. 세 모델의 절대 성능은 달랐지만 hard-gate 적용 시 Recall이 하락하는 방향은 공통적이었다. GPT 480건 3회 반복의 Binary F1 표준편차는 0.0015였다. 이러한 한계와 오류 분석 결과는 실제 SOC 데이터와 유연한 검증 구조를 이용한 후속 연구의 근거로 활용할 수 있다.

VI. 결론 및 향후 연구

본 연구는 합성·템플릿 기반 이벤트 4,800건을 대상으로 Rule-only, LLM-only, LLM+ Validator, LLM+ Sanity-Check 및 Full Pipeline을 비교하고 동일한 데이터에 GPT-4.1-mini, Claude Haiku 4.5 및 Gemini 3.1 Flash-Lite를 적용하였다. GPT-4.1-mini 주 실험에서 LLM-only는 Binary F1을 0.8152에서 0.8723으로 향상시켰으며 공격 이벤트 기반 매핑 macro-F1도 0.7048에서 0.7193으로 높였다. 반면 전체 이벤트 기반 ATT&CK macro-F1은 0.6934에서 0.6851로 소폭 감소하였다. LLM+ Sanity-Check는 FP를 333건에서 102건으로 줄였으나 Recall은 0.5711로 낮아졌고 Full Pipeline은 Precision 0.9426, Recall 0.4608, F1 0.6190을 기록하였다. Full Pipeline의 FN 1,941건은 Sanity-Check rejection 986건, LLM intrusion miss/refusal 558건, Technique disagreement 373건 및 strict evidence mismatch 24건으로 구성되었다. Claude Haiku 4.5와 Gemini 3.1 Flash-Lite의 LLM-only Binary F1은 각각 0.9065와 0.9171로 나타나 기반 모델에 따른 절대 성능 차이가 확인되었다. 그러나 세 모델 모두 hard-gate를 적용할수록 Recall이 낮아지는 공통적인 경향을 보였다. 또한 GPT-4.1-mini

반복 실험의 Binary F1은 0.8824 ± 0.0015 로 나타났다. 이상의 결과는 특정 파이프라인 전체의 우수성보다 각 구성요소가 정밀도와 재현율에 미치는 영향 및 기반 모델 의존성을 함께 고려해야 함을 보여준다. 다만 합성 데이터, 네 개의 Technique, 3:1의 클래스 비율 및 상용 폐쇄형 API 기반 모델을 사용하였다는 점에서 실제 보안관제 환경으로의 일반화에는 한계가 있다. 향후 연구에서는 실제 SOC 환경의 로그 분포와 보안장비별 특성을 반영한 데이터를 활용하여 본 연구 결과의 외적 타당성을 검증하고 Sanity-Check의 rule coverage와 ATT&CK Technique 범위를 확장할 필요가 있다. 또한 복수의 Technique 후보를 반영하는 candidate-aware 평가와 evidence normalization을 적용하고 LLM 판단, Validator 결과 및 Sanity-Check 결과를 결합하는 soft validation과 불확실한 이벤트를 분석가 검토로 전달하는 analyst fallback 구조를 구현하여 현재의 hard-gate 방식과 비교하고자 한다. 이를 통해 오탐 억제와 Recall 유지 사이의 균형을 개선하고 실제 보안관제 환경에서의 적용 가능성을 평가할 계획이다.

참고문헌

- [1] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, MITRE ATT&CK: Design and Philosophy, The MITRE Corporation, McLean: VA, Technical Report MP180360R1, 2018.
- [2] MITRE Corporation. MITRE ATT&CK Enterprise Matrix [Internet]. Available: <https://attack.mitre.org/matrices/enterprise/>.
- [3] SigmaHQ. Sigma: Generic Signature Format for SIEM Systems [Internet]. Available: <https://github.com/SigmaHQ/sigma>.
- [4] Open Information Security Foundation. Suricata User Guide: EVE JSON Output [Internet]. Available: <https://suricata.readthedocs.io/>.
- [5] Splunk. Splunk Common Information Model Overview [Internet]. Available: <https://docs.splunk.com/>.
- [6] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs Through Deep Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Dallas: TX, pp. 1285-1298, October 2017. <https://doi.org/10.1145/3133956.3134015>
- [7] H. Guo, S. Yuan, and X. Wu, "LogBERT: Log Anomaly Detection via BERT," arXiv:2103.04475, March 2021. <https://doi.org/10.48550/arXiv.2103.04475>
- [8] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep Learning for Anomaly Detection in Log Data: A Survey," *Machine Learning with Applications*, Vol. 12, 100470, June 2023. <https://doi.org/10.1016/j.mlwa.2023.100470>
- [9] M. Boffa, I. Drago, M. Mellia, L. Vassio, D. Giordano, R. Valentim, and Z. Ben Houidi, "LogPrécis: Unleashing Language Models for Automated Malicious Log Analysis," *Computers & Security*, Vol. 141, 103805, June 2024. <https://doi.org/10.1016/j.cose.2024.103805>
- [10] K. Kang, Y.-J. So, and J.-G. Park, "Pre-Trained Large Language Model Pipeline for Anomaly Detection Based on the MITRE ATT&CK Framework," *The Journal of Korean Institute of Communications and Information Sciences*, Vol. 50, No. 10, pp. 1631-1645, October 2025. <https://doi.org/10.7840/kics.2025.50.10.1631>
- [11] N. O. Jaffal, M. Alkhanafseh, and D. Mohaisen, "Large Language Models in Cybersecurity: A Survey of Applications, Vulnerabilities, and Defense Techniques," *AI*, Vol. 6, No. 9, 216, September 2025. <https://doi.org/10.3390/ai6090216>
- [12] S. Akhtar, S. Khan, and S. Parkinson, "LLM-Based Event Log Analysis Techniques: A Survey," arXiv:2502.00677, February 2025. <https://doi.org/10.48550/arXiv.2502.00677>
- [13] C. Hwang, S. Bae, and T.-J. Lee, "MITRE ATT&CK and Anomaly Detection Based Abnormal Attack Detection Technology Research," *Journal of Convergence Security*, Vol. 21, No. 3, pp. 13-23, 2021.
- [14] J. Jin, B. Tang, M. Ma, X. Liu, Y. Wang, Q. Lai, ... and C. Zhou, "Crimson: Empowering Strategic Reasoning in Cybersecurity Through Large Language Models," arXiv:2403.00878, March 2024. <https://doi.org/10.48550/arXiv.2403.00878>
- [15] R. Fieblinger, M. T. Alam, and N. Rastogi, "Actionable Cyber Threat Intelligence Using Knowledge Graphs and Large Language Models," arXiv:2407.02528, July 2024. <https://doi.org/10.48550/arXiv.2407.02528>
- [16] H. C. Nguyen, S. Tariq, M. B. Chhetri, and B. Q. Vo, "Towards Effective Identification of Attack Techniques in Cyber Threat Intelligence Reports using Large Language Models," arXiv:2505.03147, May 2025. <https://doi.org/10.48550/arXiv.2505.03147>
- [17] K. Guru, R. J. Moss, and M. J. Kochenderfer, "On Technique Identification and Threat-Actor Attribution using LLMs and Embedding Models," arXiv:2505.11547, May 2025. <https://doi.org/10.48550/arXiv.2505.11547>
- [18] R. Singh, S. Tariq, F. Jalalvand, M. B. Chhetri, S. Nepal, C. Paris, and M. Lochner, "LLMs in the SOC: An Empirical Study of Human-AI Collaboration in Security Operations Centres," arXiv:2508.18947, August 2025. <https://doi.org/10.48550/arXiv.2508.18947>

0.48550/arXiv.2508.18947

- [19] P. Jing, M. Tang, X. Shi, X. Zheng, S. Nie, S. Wu, ... and X. Luo, "SecBench: A Comprehensive Multi-Dimensional Benchmarking Dataset for LLMs in Cybersecurity," arXiv:2412.20787, December 2024. <https://doi.org/10.48550/arXiv.2412.20787>
- [20] Y. Liu, Z. Chen, S. Xu, M. He, S. Tao, W. Meng, ... and Y. Sun, "R-Log: Incentivizing Log Analysis Capability in LLMs via Reasoning-Based Reinforcement Learning," arXiv:2509.25987, September 2025. <https://doi.org/10.48550/arXiv.2509.25987>
- [21] A. M. Høst, P. Lison, and L. Moonen, "A Systematic Approach to Predict the Impact of Cybersecurity Vulnerabilities Using LLMs," arXiv:2508.18439, August 2025. <https://doi.org/10.48550/arXiv.2508.18439>
- [22] M. Bhatt, S. Chennabasappa, Y. Li, C. Nikolaidis, D. Song, S. Wan, ... and J. Saxe, "CyberSecEval 2: A Wide-Ranging Cybersecurity Evaluation Suite for Large Language Models," arXiv:2404.13161, April 2024. <https://doi.org/10.48550/arXiv.2404.13161>



김은총(Eunchong Kim)

2017년 : 호서대학교
컴퓨터소프트웨어학과
졸업(학사)

2024년 ~ 현재 : 아주대학교 정보통신대학 사이버보안학과
석사과정

※ 관심분야 : Computer Emergency Response Team
(CERT), Malware Analysis



손태식(Taeshik Shon)

2000년 : 아주대학교 정보 및
컴퓨터공학부 졸업(학사)

2002년 : 아주대학교
정보통신전문대학원
졸업(석사)

2005년 : 고려대학교 정보보호대학원
졸업(박사)

2004년 ~ 2005년 : University of Minnesota 방문연구원

2005년 ~ 2011년 : 삼성전자 통신·DMC 연구소 책임연구원

2017년 ~ 2018년 : Illinois Institute of Technology 방문교수

2011년 ~ 현재 : 아주대학교 정보통신대학 사이버보안학과
교수

※ 관심분야 : Digital Forensics, ICS/Automotive Security