클라우드 환경에서 eBPF와 LLM을 활용한 APT 탐지 프레임워크 설계*

김종섭[1] 손창민[2] 김진우[3†]

[1,2,3]광운대학교 (대학원생, 학부생, 교수)

Design of an eBPF and LLM assisted Detection Framework for APTs in Cloud Environments

Jong-Seop Kim[1], Chang-Min Son[2], Jin-Woo Kim[3]

[1,2,3]Kwangwoon University(Graduate student, Undergraduate student, Professor)

요 약

본 논문은 컨테이너 환경에서 발생하는 지능형 지속 위협(APT)을 탐지하기 위해 eBPF와 거대 언어 모델(LLM)을 결합한 새로운 프레임워크를 제안한다. 제안하는 시스템은 eBPF를 사용하여 커널 수준의 시스템 이벤트를 낮은 오버헤드로 수집하고, 이를 프로비넌스 그래프로 구조화한다. 이후 공격의 변곡점이 되는 핵심 이벤트를 중심으로 인과적으로 연결된 서브그래프를 추출하여, LLM이 이해할 수 있는 맥락 위주의 구조를 가진 프롬프트로 변환한다. LLM은 MITRE ATT&CK 프레임워크를 통해 공격의 문법과 의도를 학습함으로써, 이전에 알려지지 않은 새로운 공격 체인까지 탐지할 수 있는 잠재력을 가진다. 이 접근법은 저수준의 시스템 데이터를 통해서 고수준의 위협 인텔리전스를 탐지하는 것과 보안 전문가들에게 저수준의 로그로부터 지능형 지속 위협으로부터의 직관을 제시하는 것을 목표로 한다.

I. 서론

클라우드 네이티브 컴퓨팅 패러다임이 확산되면서, 컨테이너 기술은 현대 소프트웨어 개발 및 배포의 핵심 요소로 자리 잡았다. Docker와 Kubernetes로 대표되는 컨테이너 기술은 기존 가상 머신에 비해 월등히 가볍고 빠른 배포를 가능하게 함으로써, 마이크로서비스 아키텍처로의 전환을 가속화하고 있다.

그러나 이러한 배포의 편리성과 경량화는 새로운 보안 도전 과제를 야기했다. 다수의 컨테이너가 단일 호스트 운영체제의 커널을 공유하는 구조적 특성은 심각한 보안 위협의 공격 표면이 된다. 실제로 커널 취약점을 악용하여 컨테이너의 격리 환경을 탈출하거나 다른 컨테이너에 영향을 미치는 Cross container attack [1] 공격 기법이 보고된 바 있다. 이러한 침해 기법들은 시스템에 잠입하여 장기간 은밀하게 활동하는 지능형 지속 위협(APT, Advanced Persistent Threat)의 발판으로 사용된다.

APT 공격은 정상적인 시스템 유틸리티를 악용하고, 낮은 수준의 활동을 장기간에 걸쳐 분산시키는 전략을 사용하기 때문에 탐지가 매우 어렵다. 기존의 시그니처나 룰 기반 보안 솔루션은 이러한 개별 행위들을 악성으로 판단하기 어려우며, 전체 공격 흐름의 맥락을 파악하지 못하는 한계를 가진다.

본 논문에서는 이러한 문제점을 해결하기 위해, eBPF (extended Berkeley Packet Filter)와 거대 언어 모델(LLM)을 결합한 새로운 컨테이너 APT 공격 탐지 프레임워크를 제안한다. eBPF는 리눅스 커널을 수정하거나 모듈을 로드할 필요 없이, 샌드박스 환경에서 프로그램을 실행하여 커널의 데이터를 안전하고 효율적으로 추적할 수 있는 기술이다[2]. 이를 통해 기존 감사(audit) 시스템보다 훨씬 낮은 오버헤드로 컨테이너 내부의 시스템 호출, 파일 접근, 네트워크 연결 등 상세한 행위 데이터를 확보한다.

이후 수집된 원시 데이터를 W3C의 표준 데이터 모델인 PROV-O(Provenance Ontology)를 활용하여 구조화한다[3]. PROV-O는 데이터의 출처와 이력, 그리고 처리 과정에서의 관계를 명확하게 표현하는 온톨로지 모델로, 이를 통해 프로세스, 파일, 네트워크 소켓 간의 인과 관계를 나타내는 프로비넌스 그래프(provenance graph)를 생성한다. 정제된 그래프 데이터를 LLM이 분석 가능한 시퀀스 형태로 변환하여 입력함으로써, 모델은 단편적인 행위의 나열이 아닌 복합적인 공격 시나리오의 맥락을 학습하고 탐지하게 된다.

* 이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. RS-2024-00457937)

† 교신저자(jinwookim@kw.ac.kr)

LLM은 실제 공격 사례를 체계적으로 정리한 지식 베이스인 MITRE ATT&CK 프레임워크를 기반으로 학습한다[4]. MITRE ATT&CK은 공격자들이 사용하는 전술(tactics)과 기술(techniques)을 표준화한 모델로, 이를 학습 데이터로 활용함으로써 LLM은 알려지지 않은 변종 공격까지 식별할 수 있는 잠재력을 가지게 된다.

본 연구의 핵심적인 기여를 요약하면 다음과 같다.

- 컨테이너 환경의 프로비넌스 데이터 분석에 LLM을 도입한 새로운 접근법을 제시한다. 이를 통해 기존 룰 기반 시스템이 탐지하기 어려운 다단계 APT 공격의 문맥적(contextual) 위협을 탐지할 수 있는 가능성을 연다.
- Metasploit 프레임워크를 활용하여 실제적인 APT 공격 시나리오 기반의 개념 증명(PoC)을 수행한다. 이는 제안하는 시스템이 이론에 그치지 않고 실효성 있는 위협을 탐지할 수 있음을 실험적으로 검증한다.

II. 위협 모델

본 연구에서 제안하는 시스템이 탐지하고자 하는 위협을 명확히 정의하기 위해 다음과 같이 가정한다. 본 시스템은 공격자가 이미 웹 애플리케이션 취약점 공격 등 어떠한 수단을 통해 단일 컨테이너에 대한 초기 접근(initial access)에 성공했다고 가정한다. 또한 본 연구의 탐지 범위는 침입 시도가 아닌, 침입 이후에 발생하는 악의적인 내부 활동에 초점을 맞춘다. 공격자의 최종 목표는 APT 공격의 일반적인 특성을 따라, 중요 정보를 탈취하기 위한 데이터 유출(data exfiltration)로 설정한다.

이를 위해 공격자는 시스템에 대한 이해도를 높이고, 권한을 확장하며, 최종적으로 데이터를 외부로 빼돌리기 위한 일련의 과정을 수행한다. 공격자는 목표 달성을 위해 시스템에 기본적으로 내장된 도구들을 악용하는 전술을 적극적으로 활용한다. 본 연구에서 탐지하고자 하는 공격자의 주요 전술 및 기술(TTPs, Tactics, Techniques, and Procedures)은 MITRE ATT&CK 프레임워크를 기반으로 하며, 다음을 포함한다.

- **실행 (Execution)**: bash와 같은 셸을 통해 시스템에 임의의 명령어를 실행한다.
- **지속성 (Persistence)**: cron 작업이나 .bashrc와 같은 시작 스크립트를 수정하여 시스템 재부팅 이후에도 지속적으로 실행될 수 있는 백도어를 마련한다.
- **방어 회피 (Defense Evasion)**: 감사 로그(audit logs), 셸 히스토리(.bash_history) 등 자신의 흔적을 삭제하여 탐지를 회피한다.
- **자격 증명 접근 (Credential Access)**: 환경 변수, 소스 코드, 설정 파일 등에 노출된 API 키나 비밀번호, SSH 키와 같은 자격 증명을 수집한다.
- **탐색 (Discovery)**: ps, netstat, find 등의 명령어를 사용하여 시스템의 프로세스, 네트워크 연결, 파일 시스템을 정찰하고 공격에 유용한 정보를 수집한다
- **측면 이동 (Lateral Movement)**: 탈취한 자격 증명을 이용해 ssh 등으로 클러스터 내의 다른 컨테이너나 내부망의 다른 호스트로 접근을 시도한다.
- **유출 (Exfiltration)**: curl이나 wget을 사용한 HTTP/HTTPS 통신, 또는 DNS 터널링과 같은 비정상적인 프로토콜을 이용하여 수집한 데이터를 외부 C&C(Command and Control) 서버로 전송한다.

본 연구는 초기 침투 자체를 방지하는 것과 eBPF 에이전트 자체를 무력화하는 커널 레벨의 공격, 시스템의 가용성을 침해하는 대규모 DoS/DDoS 공격 그리고 LLM 모델 자체를 기만하기 위한 적대적 공격은 탐지 범위에 포함하지 않는다.

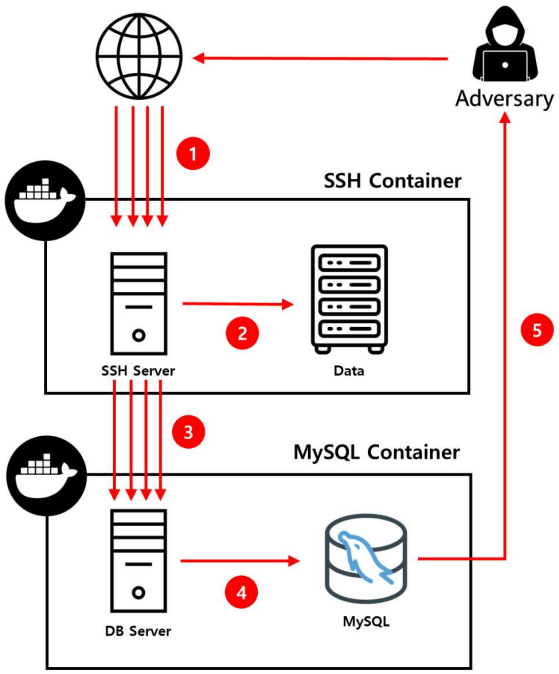

Figure 1. An APT scenario in a cloud environment where SSH and MySQL containers reside

III. 시나리오

본 연구에서 제안하는 시스템의 탐지 성능을 검증하기 위해, 다단계로 구성된 실제적인 공격 시나리오를 설계한다. Fig. 1는 공격자가 초기 접근에 성공한 컨테이너를 발판 삼아 내부망의 다른 컨테이너로 수평 이동하여 최종적으로 데이터를 유출하는 과정을 포함한다. 공격의 전체 흐름은 다음과 같다.

1. 공격자는 외부에 공개된 SSH 컨테이너에 대해 무차별 대입 공격(Brute-force Attack)을 수행하여 시스템에 대한 내부 쉘 권한을 획득한다.

2. 침투에 성공한 공격자는 SSH 컨테이너 내부에서 쉘 히스토리(.bash_history)와 같은 파일을 조사하여, 내부망에 존재하는 다른 MySQL-DB 컨테이너와의 통신 기록 및 자격 증명 정보를 확인한다.

3. 공격자는 내부 정찰을 통해 획득한 자격 증명(Username/Password)을 이용해 MySQL-DB 컨테이너의 데이터베이스 서버로 접근을 시도한다.

4. 데이터베이스 접근에 성공한 공격자는, 획득한 권한으로 MySQL 내부를 조사하며 탈취할 만한 중요 데이터를 물색한다.

5. 공격자는 최종적으로 내부 테이블 조회 등을 통해 수집한 중요 정보를 외부의 C&C(Command and Control) 서버로 전송하여 데이터를 탈취한다.

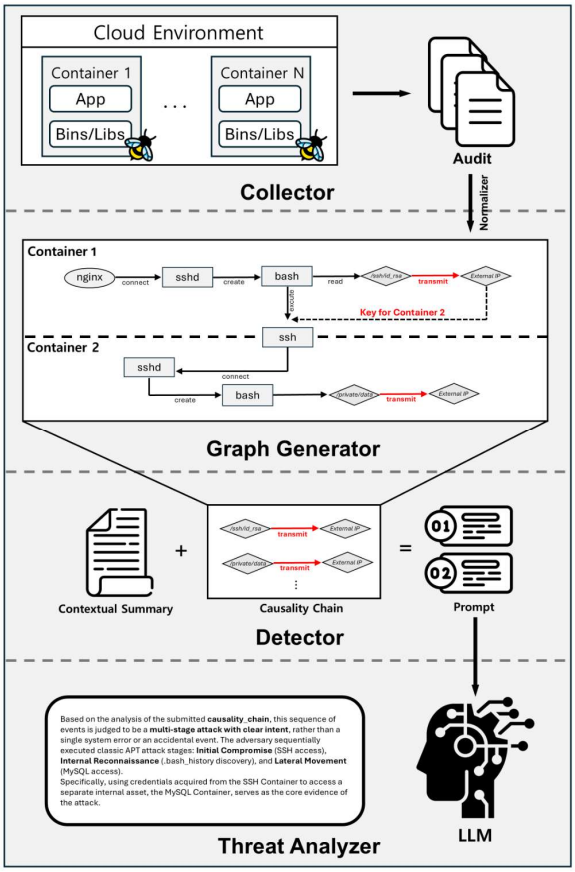

Figure 2. System overview

IV. 시스템 개요

본 연구에서 제안하는 컨테이너 APT 공격 탐지 시스템은 4단계의 파이프라인 아키텍처로 구성된다.

각 컴포넌트는 커널에서 수집된 저수준의 원시 데이터를 점진적으로 정제하고 분석하여, 최종적으로 인간 분석가가 이해할 수 있는 고수준의 위협 인텔리전스로 변환하는 역할을 수행한다.

Collector는 각 호스트에 배포되어 eBPF를 이용해 커널 레벨의 시스템 행위 데이터를 수집한다. 커널을 직접 수정하지 않고 KProbes와 Tracepoints와 같은 eBPF 기술을 활용하여, 운영 환경의 성능에 미치는 영향을 최소화하면서도 필요한 데이터를 확보한다. 위협 모델에서 정의한 TTPs를 효과적으로 탐지하기 위해, Collector는 프로세스 활동(execve, fork), 파일 시스템 접근(openat, unlinkat), 네트워크 통신(socket, connect), 권한 관리(setuid) 등과 관련된 핵심 시스템 호출들을 중점적으로 추적한다.

Graph Generator는 수집된 원시 로그를 표준화된 이벤트 튜플로 변환하고, 이를 입력받아 시스템 행위의 인과 관계를 나타내는 프로비넌스 그래프(Provenance Graph)를 구축한다. 이 그래프는 W3C의 PROV-O 온톨로지 모델을 기반으로 하며, 시스템의 구성 요소를 프로세스는 Agent, 파일/소켓은 Entity, 시스템 호출은 Activity로 매핑한다. Fig. 2에서 주체의 행위는 Edge로 표현되며, 최초 시작점이 되는 프로세스는 원, 행위의 주체인 일반적인 프로세스는 직사각형, 데이터의 흐름에 대한 목표물은 마름모로 표현된다. 예를 들어, nginx 프로세스가 execve를 통해 bash를 실행했다면, nginx(최초 시작점)가 execve(Edge)를 통해 bash(프로세스)를 생성했음을 나타내는 노드와 엣지가 그래프에 추가된다. 이 과정을 통해 시스템에서 발생한 이벤트의 선후 관계와 상호작용이 구조화된 그래프 형태로 누적된다.

Detector는 시스템의 핵심 두뇌로써 execve, connect 등 공격의 흐름에서 변곡점이 되는 핵심 이벤트가 발생하면, 해당 이벤트를 중심으로 인과적으로 연결된 주변 노드와 엣지를 탐색하여 관련된 행위의 묶음인 서브그래프(sub-graph)를 추출한다. 이후, 추출된 서브그래프는 LLM이 가장 잘 이해할 수 있는 형태의 구조화된 JSON 프롬프트로 변환된다. 이 프롬프트는 LLM에게 명확한 분석을 요청하기 위해 공격의 배경을 설명하는 행위 요약(contextual_summary), 구체적인 증거 목록인 인과관계 설정(causality_chain), 그리고 명확한 질문인 query 필드를 포함한다.

Threat Analyzer는 Detector 로부터 구조화된 프롬프트를 전달받는다. 이 컴포넌트는 내부적으로 사전 학습된 거대 언어 모델(LLM)을 활용하여, 주어진 contextual_summary와 causality_chain을 바탕으로 해당 행위 시퀀스의 위험도를 종합적으로 평가한다. 해당 분석 과정을 통해 행위를 MITRE ATT&CK TTPs과 매핑하고, 위협으로 판단된 근거를 자연어로 설명한다. 생성한 최종 분석 결과는 보안 분석가가 복잡한 로그를 직접 분석하는 시간을 줄이며, 위협의 심각성 및 종류를 신속하게 판단할 수 있게 하여 즉각적 대응을 가능하게 한다.

V. 관련 연구

Falco [5]는 eBPF를 이용해 시스템 이벤트를 실시간으로 감시하고 사전에 정의된 규칙(rule) 집합에 기반하여 악성 행위를 탐지한다. Falco는 알려진 위협 시나리오를 탐지하는 데 유용하지만, 규칙 기반 접근 방식의 본질적인 한계로 인해 여러 단계를 거치며 정상 행위를 위장하는 APT 공격의 전체적인 공격 체인을 파악하는 데는 어려움이 있다.

또한 Cilium [6]는 eBPF를 활용하는 대표적인 오픈소스 프로젝트로, 컨테이너 환경의 네트워킹, 관찰 가능성, 그리고 보안을 강화하는 데 중점을 둔다. Cilium은 eBPF를 통해 커널 레벨에서 네트워크 정책을 강제하고 API 레벨의 통신까지 깊이 있게 분석할 수 있는 강력한 기능을 제공한다. 그러나 Cilium의 핵심은 네트워크 보안 정책 강제에 있으며, 본 연구가 목표로 하는 프로세스, 파일 접근, 네트워크 행위를 종합하여 공격의 전체적인 맥락을 이해하고 탐지하는 것과는 다른 접근 방식을 가진다.

SLEUTH [7]는 실시간으로 시스템 전체의 프로비넌스 그래프를 구축하고 의심스러운 객체에 태그를 전파하여 스텔스 공격을 탐지하는 기법을 제안했다. 후속 연구인 HOLMES [8]는 정보 이론을 적용하여 수많은 이벤트 흐름 중에서 의심스러운 공격 시나리오를 자동으로 식별하고 순위를 매기는 방법을 제시했다. 이 연구들은 APT 탐지 자동화의 기틀을 마련했으나, 그 결과물은 분석가가 해석해야 하는 의심 점수나 태그된 그래프의 형태였다. 또한, 사전에 정의된 공격 템플릿을 벗어나는 새로운 형태의 공격 서사를 이해하는 데는 여전히 한계가 있었다.

따라서 본 연구는 기존 연구들의 성과를 바탕으로, LLM을 활용하여 컨테이너 프로비넌스 데이터를 의미론적으로 해석하고, 이를 통해 실제 공격 위협의 전체 체인을 자동으로 탐지하는 새로운 방법을 제안한다.

VI. 실험 및 평가 방안

시스템 구현을 위해 libbpf를 사용하여 execve, connect등의 사전에 정의한 system call을 추적하며, (Agent, Activity, Entity) tuple로 구성해 provenance graph를 구성한다. 그래프 탐색을 통해 인과적 그래프를 추출하여 JSON 프롬프트로 변환하며, 상용 LLM API와 연동하여 프롬프트를 분석하고 TTPs 매핑 및 위험도 평가를 수행하도록 구현한다. 실험 평가는 Docker container 기반 테스트베드를 구축하며, Metasploit 프레임워크를 활용하여 TTPs 기반의 APT 공격 시나리오 데이터를 생성하여 수행한다[9].

탐지 성능은 정확도, 정밀도, 재현율, F1-Score와 같은 표준 분류 지표와 오탐률 및 미탐률을 통해 종합적으로 평가된다. 제안 시스템의 우수성을 입증하기 위해, 대표적인 규칙 기반 보안 도구인 Falco와의 비교 평가를 수행한다. 특히 Falco의 기본 규칙으로는 탐지하기 어려운 다단계 APT 공격 시나리오에 대해 제안 시스템이 더 높은 탐지율을 보이는지 집중적으로 분석하여, LLM 기반의 의미론적 분석이 갖는 차별화된 강점을 실험적으로 입증하고자 한다.

VII. 결론

본 논문은 eBPF와 거대 언어 모델(LLM)을 결합하여 컨테이너 환경의 지능형 지속 위협(APT)을 탐지하는 새로운 프레임워크를 제안했다. eBPF를 통해 시스템 행위를 낮은 오버헤드로 수집하고, 이를 프로비넌스 그래프로 변환한 뒤, LLM의 강력한 문맥 이해 능력을 활용하여 알려지지 않은 공격 체인까지 식별할 수 있는 가능성을 보였다. 이 접근법은 저수준의 시스템 이벤트를 보안 분석가가 직관적으로 이해할 수 있는 고수준의 위협 인텔리전스로 자동 변환하는 목적도 함께 가진다.

연구에서는 본 논문에서 제안하는 시스템의 실효성과 성능을 체계적으로 검증한다. 이를 위해 실제 운영 환경과 유사하게 가상화 플랫폼인 Proxmox 상에 구축된 Ubuntu VM 환경에 Docker 기반의 마이크로서비스 애플리케이션을 배포하여 실험을 진행한다. 공격 데이터셋은 Metasploit 프레임워크를 활용하여, 초기 침투, 내부 정찰, 데이터 유출 등 다단계 TTPs 기반의 공격 시나리오를 실제로 실행시켜 수집한다.

[참고문헌]

[1] N. Papastergiou, K. Mallas, and V. P. Kemerlis, “Cross-Container Attacks: The Bewildered eBPF on the Hunt for the Vile Namespace,”in 31st USENIX Security Symposium (USENIX Security 22). USENIX Association, 2022, pp. 257-274

[2] "eBPF," eBPF.io/

[3] "PROV-O," http://www.w3.org/TR/prov-o/

[4] "MITRE ATT&CK," https://attack.mitre.org/

[5] "Falco," https://falco.org/

[6] "Cilium," https://cilium.io/

[7] M.N. Hossain, S. M. Milajerdi, R. Gjomemo, Z. Saw-h, A. Bates, T. Skandier, C. Nita-rotaru, and M. Bailey, “SLEUTH: Real-time attack scenario reconstruction from COTS audit data,” in 28th USENIX Security Symposium (USENIX Security 19), 2019.

[8] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. N. Venkatakrishnan, "HOLMES: Real-time APT detection through correlation of suspicious information flows," in 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019, pp. 1137-1152.

[9] "Metasploit," https://www.metasploit.com/