δ-Tracer: 도커 이미지 델타 추적을 통한 위협벡터 탐색 프레임워크

우상민*, 이재욱*, 황대성*, 장도원*, 최상훈[1], 박기웅†

*세종대학교 SysCore Lab. (학부생, 연구교수, 교수)

δ-Tracer: Threat Vector Detection Framework with Docker Image Delta Tracing

Sang-Min Woo*, Jae-Wook Lee*, Dae-Sung Hwang*, Do-Won Jang*, Ki-Woong Park†, Sang-Hoon Choi[1]

*SysCore Lab., Sejong University

**Dept. of Computer and Information Security, Sejong University

(Graduate Student, Research Professor, Professor)

요 약

클라우드 환경은 빠르고 반복적인 DevOps 기반 개발·배포로 인해 구성 요소가 동적으로 변화하며, 이에 따라 보안 취약점의 추적 및 대응이 점점 더 어려워지고 있다. 특히 패치나 설정 변경과 같은 정상적인 시스템 업데이트가 오히려 새로운 공격 벡터를 유발할 가능성이 존재한다. 이에 본 논문에서는 도커 기반 클라우드 환경에서 서로 다른 시점의 이미지를 비교·분석하여 변경 사항을 도출한다. 그리고 이러한 변경 사항이 보안 취약점으로 연결될 가능성을 MITRE ATT&CK 프레임워크 기반의 TTP 시나리오를 통해 분석하는 국내 보안 진단 플랫폼을 제안한다. 본 플랫폼은 지속적으로 누적되는 클라우드 환경의 변경 이력 속에서 실제 위협으로 이어질 수 있는 변화를 식별함으로써 복잡한 클라우드 운영 환경에서의 보안 분석 효율성과 실용성을 제공할 수 있다.

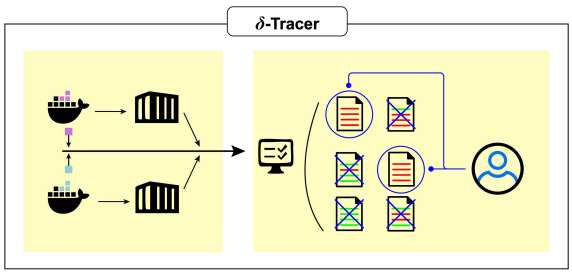
I. 서론

전세계적인 디지털 전환의 가속화에 발맞추어 국내에서도 클라우드 마이그레이션이 활발히 진행되고 있다. 또한 빠르고 반복적인 개발을 위해 많은 조직은 DevOps 기반 개발 방식을 채택하고 있다. 이러한 환경에서는 시스템 구성과 소프트웨어 상태가 지속적으로 변화하기 때문에, 보안 관리와 위협 분석이 더욱 복잡해진다. 특히 여러 패치에 걸쳐 발생한 백업 및 로그 기록 미삭제, 환경 변수 설정 오류, 불필요한 퍼블릭 포트 개방 등과 같이 시스템 자체의 결함이 아닌 사용자에 의해 유발되는 보안 취약점들은 클라우드 네이티브 보안만으로 추적하기 어렵다. 2023년 Microsoft Azure DevOps Server에서 발견된 취약점 사례는 패치가 오히려 새로운 공격 지점을 유발할 수 있음을 보여준다 [1]. 이와 더불어 점점 고도화·자동화되는 공격 속도에 맞추어 클라우드 보안 관리는 자동화와 오케스트레이션의 적극적 활용이 권장되고 있다 [2]. 해외의 많은 클라우드 보안 소프트웨어에서는 앞서 언급한 내용을 반영하고 있다. 예를 들어, AWS CloudFormation은 인프라 코드와 리소스 간 변경을 자동으로 탐지하는 드리프트 감지 기능을 제공한다 [3]. 또한 Microsoft Defender for Containers는 실행 중인 컨테이너 내 바이너리 변경을 탐지하는 바이너리 드리프트 감지 기능을 지원한다 [4]. 하지만 이러한 보안 도구들은 이미지와 이미지로부터 생성한 서로 다른 컨테이너를 정, 동적으로 분석하는 기능을 제공하지

† 교신저자: 박기웅 (세종대학교 정보보호학과 교수)

않는다. 또한 국내 클라우드 벤더에서도 많은 부분을 보완하고 있지만 앞서 언급한 드리프트 탐지 기능을 비롯해 아직 지원되지 않는 보안 기능들이 다수 존재한다.

이에 본 논문에서는 서로 다른 시점의 클라우드 이미지를 비교 분석하고, 이미지 간 변화로 인해 발생할 수 있는 취약 요소를 탐지해 공격 시나리오와 연결 지어 사용자에게 제공하는 국내 보안 진단 플랫폼을 제안하고자 한다. 플랫폼 시스템 구성을 표현한 그림은 (그림 1)과 같다. 본 논문은 2장 δ-Tracer 디자인 및 프로토타입, 3장 결론 및 추후 연구, 참고 문헌으로 구성된다.



(그림 1) δ-Tracer 시스템 그림

II. δ-Tracer 디자인 및 프로토타입

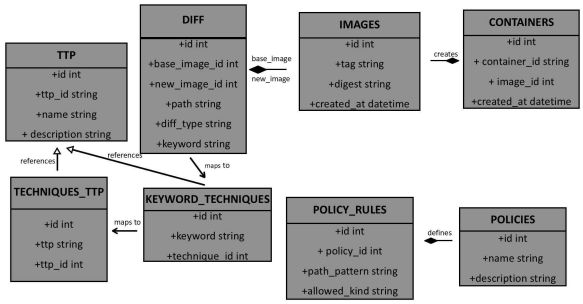
본 논문은 도커 기반 클라우드 환경에서 패치로 인한 보안 취약점의 원인 요소 파악 문제를 해결하기 위한 플랫폼을 제안하는 초기 연구 단계를 서술한다.

3.1 δ-Tracer는 MITRE ATT&CK을 기반으로 취약점을 분류한다. MITRE ATT&CK은 국방부(DOD), 사이버보안 및 인프라 보안국(CISA), 국가표준기술연구소(NIST) 등 권위 있는 기관과 협력을 통해 실제 사이버 공격 사례를 기반으로 공격자의 전술과 기술을 체계적으로 정리한 지식 기반 프레임워크다 [5]. 본 플랫폼은 취약점 분류 모델로서 MITRE ATT&CK for Cloud 매트릭스를 활용한다. 매트릭스는 공격자의 달성 목표인 전술이 행을, 전술의 실현을 위한 공격기법이 각 항목을 구성한다. TTP(Tactics, Techniques, and Procedures)는 이러한 전술과 기술을 바탕으로 하는 공격 시나리오 전반을 의미한다.

3.2 δ-Tracer는 IaC를 통해 인프라를 관리하는 서버를 대상으로 분석을 수행한다. 분석은 서버의 여러 시점 중 두 시점을 골라 비교하는 방식으로 진행된다. 분석 후 추출된 취약점은 MITRE ATT&CK의 매트릭스에 이벤트를 기반으로 분류된다. 분석과 취약점 분류를 위해 여러 도구를 활용한다. 대표적으로 Falco는 컨테이너 런타임 시 발생하는 시스템 콜, 파일 및 네트워크 활동을 실시간으로 모니터링하여 보안 규칙에 위반되는 행위를 탐지한다 [6].

3.3 δ-Tracer의 시스템은 시점 선택, δ-데이터 추출, δ-데이터 기반 취약점 분석, 결과 제공의 4단계로 구성된다. 먼저 시전 선택 단계에서 사용자는 서버에게 클라우드 환경에서 사용중인 Git 기반 IaC 레지스트리의 정보와 클라우드 API 접근 키를 제공한다. 서버는 Git API를 활용해 사용자의 클라우드 인프라 정보를 가져오고, 클라우드 API를 활용해 스토리지 스냅샷 등 사용자 환경에서 IaC를 기반으로 서버를 생성하기 위한 실질 요소를 획득한다. 서버는 사용자에게 IaC 레지스트리의 커밋 리스트를 타임라인에 따라 제공한다. 사용자는 제공된 정보를 바탕으로 비교하고자 하는 두 시점 A, B를 선택한다. δ-데이터 추출 단계에서 서버는 시점 A가 B로 변화하기 위해 발생한 명령어, 파일 수정을 정, 동적으로 추출하는 단계이다. 우선 선택된 두 시점에 대해 API를 통해 얻은 정보를 바탕으로 서버를 생성한 후 정, 동적 분석을 진행한다. 정적 분석 단계에서는 파일 비교 도구를 활용해 두 시점의 IaC 간 차이점을, 스토리지 비교 도구를 활용하여 파일 시스템 간 차이를 추출한다. 이 과정에서 추출된 데이터를 바탕으로 동적 분석을 진행한다. 동적 분석 단계에서 서버는 시점 A가 B로 변화하기 위해 실행된 명령어와 수정된 파일의 내용을 추출한다. 명령어는 수정된 파일의 종류에 따라 미리 준비한 플레이북을 실행해 추출해낸다. 플레이북은 특정 파일 내에서 예상 가능한 변경 요소를 만들기 위한 명령어의 집합이다. 예를 들어 만약 A에서 B로 변하며 /etc/passwd에서 내용이 추가되었음이 확인되면 플레이북을 실행함

으로써 adduser 명령어가 실행되었음을 확인할 수 있다. δ-데이터 기반 취약점 분석 단계에서 서버는 동적 분석 단계에서 추출한 명령어 실행에 의한 이벤트를 Falco 등의 규칙을 활용하여 MITRE ATT&CK 매트릭스 TTP와 연결한다. 마지막으로 결과 제공 단계에서 서버는 정, 동적 분석을 통해 얻어낸 δ-데이터와 연결된 TTP에 대한 정보를 대시보드를 통해 사용자에게 제공한다. 변경 사항과 TTP 간 연결을 위한 데이터베이스의 구성은 (그림2)와 같다. 플랫폼은 DIFF의 keyword가 TTP로 연결될 경우 시스템이 취약하다고 판단한다.



(그림 2) 데이터베이스 모델링 다이어그램

만약 시스템 내 유효한 공격 시나리오가 검출되면 서버는 해당하는 시나리오를 사용자에게 리포트의 형태로 제공한다.

II. 결론 및 추후 연구

본 연구에서는 클라우드 환경에서 서로 다른 시점의 시스템 변화를 분석해 보안 취약점을 탐지하고, 변경 사항이 MITRE ATT&CK 기반 TTP에 해당하는지 제공하는 보안 진단 플랫폼을 제안하였다. 본 플랫폼은 패치와 시스템 변경으로부터 어떤 변화가 실제 위협과 연결될 수 있는지를 TTP 기반으로 판단할 수 있는 구조를 제공한다는 점에서, 복잡한 클라우드 운영 환경에서의 보안 진단 효율성을 높일 수 있다.

추후 본격적인 개발을 통한 시스템 구현의 구체화, TTP 분류를 위한 정교한 방법론 적용, 대시보드를 통한 데이터 시각화가 필요하다. 이외 드리프트 검출을 비롯한 런타임 변화 요소 추적 분석 기능, 애플리케이션 수준의 정·동적 분석 기능 등의 추가를 고려할 수 있다.

[참고문헌]

[1] R. Ravindran, “Azure Pipelines vulnerability spotlights supply chain threats,” *TechTarget*, Feb. 1, 2023. [Online]. Available: https://www.techtarget.com/searchsecurity/news/365534205/Azure-Pipelines-vulnerability-spotlights-supply-chain-threats (accessed May 1, 2025)

[2] National Security Agency (NSA), *Advancing Zero Trust Maturity Throughout the Automation and Orchestration Pillar*, U.S. Department of Defense, Jul. 2024. [Online]. Available: https://media.defense.gov/2024/Jul/10/2003500250/-1/-1/0/CSI-ZT-AUTOMATION-ORCHESTRATION-PILLAR.PDF (accessed May 1, 2025)

[3] Amazon Web Services, “AWS CloudFormation 드리프트 감지 기능 출시,” *AWS Korea Blog*, Nov. 17, 2018. [Online]. Available: https://aws.amazon.com/ko/blogs/korea/new-cloudformation-drift-detection/ (accessed May 5, 2025)

[4] Microsoft, “Binary drift detection - Microsoft Defender for Containers,” *Microsoft Docs*, 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/defender-for-cloud/binary-drift-detection (accessed May 5, 2025)

[5] MITRE Corporation, *MITRE ATT&CK®: Design and Philosophy*, Mar. 2020. [Online]. Available: https://attack.mitre.org/docs/ATTACK_Des

ign_and_Philosophy_March_2020.pdf (accessed May 7, 2025)

[6] Falcosecurity, “Falco: Cloud Native Runtime Security for Containers,” GitHub repository, 2018. [Online]. Available: https://github.com/falcosecurity/falco (accessed May 7, 2025)