클라우드 네이티브 환경에서의 보안 위협 분석 및 대응 방안 제안

라하연[1], 엄익채[2]
[1]전남대학교 정보보안융합학과 석사과정
[2]전남대학교 정보보안융합학과 교수
kisa.rahy@gmail.com, iceuom@chonnam.ac.kr

Security Threat analysis and response plan proposal in cloud native environment

Ha-Yeon Ra[1], Leck-Chae Euom[2]
[1]Dept. of Information Security Convergence, Chonnam National University
[2]Dept. of Information Security Convergence, Chonnam National University

요 약

클라우드 네이티브 기술은 높은 민첩성과 확장성을 기반으로 다양한 산업 분야에서 빠르게 도입되고 있다. 컨테이너, 오케스트레이션, 마이크로서비스, 서버리스 등으로 구성된 이 환경은 효율적인 서비스 운영을 가능하게 하지만 그만큼 보안위협도 다양하고 복잡해진다. 본 논문에서는 클라우드 네이티브 환경에서 발생 가능한 보안 위협을 컨테이너, 오케스트레이션, 네트워크, 서비스 간 통신, 서버리스, 공급망 보안, 클라우드 인프라 등의 범주로 구분하고 이에 대한 실용적인 대응 방안을 제시한다. 이를 통해 클라우드 네이티브 환경에서의 보안성 강화를 위한 방향성을 제안한다.

1. 서론

디지털 전환이 가속화되면서 클라우드 네이티브 기술은 민첩하고 확장가능한 서비스 개발 및 운영의 핵심 요소로 부상하고 있다[1]. 컨테이너, 마이크로서비스, 오케스트레이션, 서버리스, DevOps 자동화 등은 높은 유연성과 빠른 배포를 지원한다. 이러한 복합적이고 동적인 환경에서는 전통적인 경계 기반 보안체계만으로 다양한 위협에 효과적으로 대응하기 어렵다. 특히 <표 1>과 같이 기존 애플리케이션과 클라우드 네이티브 애플리케이션은 아키텍처, 실행환경, 운영방식, 보안모델 등에서 근본적인 차이를 가지며 이에 따라 새로운 보안 접근 방식이 요구된다.

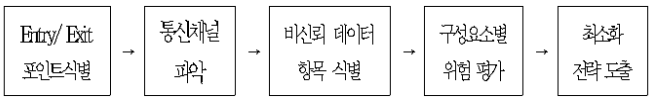
<표 1> 기존 애플리케이션과 클라우드 네이티브 애플리케이션의 차이점[2]

구분	기존 애플리케이션	클라우드 네이티브 애플리케이션
구조	모놀리식(Monolithic)	마이크로서비스
실행환경	물리 서버 기반	가상 컨테이너 기반
인프라 의존성	인프라 종속적	인프라 독립적, 이식성 보장
빌드·배포	수작업, 긴 시간	CI/CD, 자동화, 짧은 시간
조직구조	단절된 개발, 운영, 보안팀	DevOps 협업
보안모델	경계 기반 보안, 방화벽·VPN 중심	제로트러스트 보안, 워크로드 기반 접근제어

이에 본 논문은 클라우드 네이티브 환경을 구성하는 주요 기술 요소를 컨테이너, 오케스트레이션, 네트워크, 마이크로서비스 간 통신, 서버리스 및 자동화, 공급망 보안, 클라우드 인프라 일곱 가지 범주로 분류하고 요소별로 발생할 수 있는 보안 위협을 Attack Surface 관점에서 분석하여 실무 적용이 가능한 대응 방안을 제시하고자 한다.

2. Attack Surface Management 방법론 적용

Attack Surface란 공격자가 시스템에 침입하기 위해 이용할 수 있는 모든 접점(Entry/Exit Point), 통신채널, 데이터 항목을 의미한다. Mandahate와 Wing[3]은 [그림 1]과 같이 Attack Surface를 체계적으로 식별하고 수량화함으로써 시스템 보안성을 평가하고, 공격 가능 지점을 줄여나가는 것을 제안하였다.



(그림 1) Attack Surface Management 절차

클라우드 네이티브 환경에 적용하여 주요 기술 요소별 보안 위협을 다음과 같은 절차로 분석하였다. 먼저, 클라우드 네이티브 환경의 구성 요소인 컨테

이너, 오케스트레이션, 네트워크, 서비스 간 통신, 서버리스, 공급망, 클라우드 인프라를 기준으로 외부 및 내부 위협자가 접근할 수 있는 인터페이스를 식별하여 Entry/Exit Point를 파악하였다. 다음으로, 마이크로서비스 간 API 통신, 서버리스 이벤트 호출, 클러스터 관리 포트를 통한 네트워크 통신 흐름을 분석하고 암호화 및 인증 적용 여부를 평가함으로써 통신채널의 보안성을 점검하였다. 또한, API 입력값, 서버리스 이벤트, 외부 사용자 요청 등 신뢰 되지 않은 데이터 항목을 식별하여 공격자가 조작할 수 있는 경로를 확인했다. 이후, 각 기술 요소별로 식별된 Attack Surface를 기반으로 주요 보안 위협을 도출하였다. 마지막으로, 분석 결과를 바탕으로 최소 권한 적용, 무결성 검증, 통신 암호화, 입력 검증 및 필터링, 가시성 및 모니터링 강화, 공급망 보안 관리 등 공통 대응 방안을 수립하여 클라우드 네이티브 환경의 보안 위협을 최소화하는 방향을 제시하고자 한다.

3. 클라우드 네이티브 보안 위협에 따른 대응 방안

가. 클라우드 네이티브 보안 위협 분석

1) 컨테이너 보안 위협 (A)

컨테이너는 애플리케이션과 그 실행 환경(라이브러리, 설정 파일 등)을 하나의 패키지로 묶어 실행할 수 있도록 만든 기술로 애플리케이션을 경량화하고 빠르게 배포할 수 있게 한다. 컨테이너 런타임 환경의 격리 실패로 인해 컨테이너 탈출(Container Escape)이 발생하여 그 시스템의 자원이나 다른 컨테이너에 비정상적으로 접근하거나 제어하는 위협이 발생할 수 있다(a1). 또한, 보안 검증이 불충분한 컨테이너 이미지를 사용하여 애플리케이션을 배포할 경우 악성 코드 삽입 위험이 존재한다(a2).

2) 오케스트레이션 기반 보안 위협 (B)

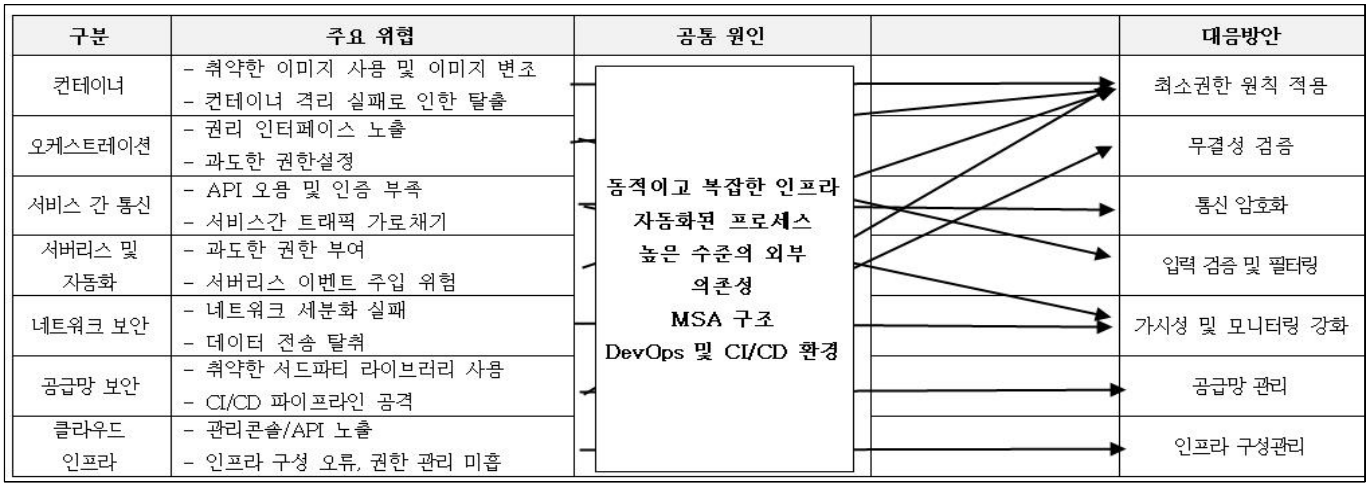
쿠버네티스(Kubernetes)와 같은 오케스트레이션 도구는 컨테이너 관리 자동화를 지원할 수 있다. 그러나, 사용자, 서비스 계정, 애플리케이션에 과도한 권한 부여 또는 미흡한 인증 관리로 인한 보안 취약점을 초래할 수 있다[5, 8, 9](b1). 쿠버네티스 대시보드는 클러스터 상태를 시각적으로 확인하고 관리할 수 있는 인터페이스인데, 이러한 대시보드가 인터넷에 직접 노출되어 무단 접근이 가능할 경우 클러스터 제어나 민감정보 유출 경로로 악용될 수 있다. (b2)

3) 서비스 간 통신 보안 위협 (C)

클라우드 네이티브 애플리케이션은 API와 마이크로서비스, 서비스 메시 등을 통해 서로 통신한다. API를 의도하지 않는 방식으로 사용하거나, API에 접근하는 사용자나 시스템의 신원을 적절히 확인하지 않으면 보안 위협이 발생할 수 있다[4](c1). 서비스 간 트래픽 가로채기는 마이크로서비스 사이에 오가는 네트워크 통신을 무단으로 감시하거나 조작하는 행위로, 중간자 공격(Man-in-the Middle Attack)의 주요 원인이 된다(c2).

4) 서버리스 및 자동화 보안 위협 (D)

서버리스는 개발자가 인프라를 관리하지 않고 코드만 업로드하면 자동으로 실행되는 실행환경으로 빠른 확장성과 유연성을 제공한다. 또한, CI/CD 자동화 파이프라인과 결합하면서 개발에서 배포까지의 프로세스가 빠르게 전환되는 DevOps 생태계와 밀접하게 연동된다. 함수나 서비스에 필요 이상의 권한이 부여되거나 부적절하게 관리되면 보안 위협이 발생한다(d1). 또한, 서버리스 함수의 경우 이벤트 기반 구조 특성상 데이터를 조작하여 의도하지 않은 동작을 유발하는 등 보안 위협이 존재한다(d2).



(그림 2) 클라우드 네이티브 환경에서의 보안 위협 및 대응 방안

5) 네트워크 보안 위협 (E)

클라우드 네이티브 환경의 네트워크는 복잡한 구조를 가져 가상네트워크, 서브넷, 보안그룹, 네트워크 정책 등을 사용하여 워크로드와 서비스를 논리적으로 분리하여 서비스 간 네트워크를 세분화한다. 이러한 네트워크 분리나 설정이 부적절하게 이루어지거나, 세분된 환경 간 통신이 올바르게 제어하지 않을 경우 문제가 발생할 수 있다(e1). 네트워크 통신 간 데이터 암호화 부재로 인해 공격자가 내부 네트워크를 가로채거나 암호화되지 않은 데이터를 탈취할 수 있다[5](e2).

6) 공급망 보안 위협 (F)

공급망 보안 위협은 많은 오픈소스 소프트웨어, 외부 라이브러리 등을 사용하기 때문에 위협 요소가 존재한다. 특히 취약한 서드파티 라이브러리를 통해 악성 코드가 삽입되거나, (f1) 빌드 파이프라인이 공격받으면 전체 시스템의 신뢰성과 보안이 심각하게 위협받을 수 있다[6](f2).

7) 클라우드 인프라 보안 위협 (G)

클라우드 네이티브는 기본적으로 퍼블릭 클라우드 또는 하이브리드 클라우드 인프라 위에서 구동된다. 이 인프라 계층은 네트워크, 스토리지, 가상머신(VM), 클라우드 관리 콘솔 등 다양한 자원으로 구성되며 이들에 대한 설정 오류, 권한관리 미흡 등 위협이 존재한다. 클라우드 리소스에 대한 권한을 사용자 또는 서비스 계정에 불필요하게 부여한 경우 보안사고 발생 시 피해 범위가 확대된다(g1). 또한, 가상 자원의 설정 오류로 인해 민감한 데이터가 공개되는 경우도 존재한다(g2).

나. 클라우드 네이티브 보안 대응 방안

클라우드 네이티브 환경에서 식별된 보안 위협에 대응하기 위해 본 논문은 <표 2>와 같이 공통적으로 적용할 수 있는 대응 방안을 정리하였다.

<표 2> 클라우드 네이티브 보안 대응 방안

대응 방안	설명	관련 위협(코드)
최소 권한 원칙 적용	사용자, 서비스, 애플리케이션에 필요한 최소 권한 만 부여	a1, a2, b1, c1, d1, e1, f1
무결성 검증	컨테이너 이미지, 코드, 아티팩트에 대한 서명 및 무결성 검증 수행	a1, a2, f1, f2
통신 암호화	서비스 간 통신 및 데이터 전송 시 TLS/mTLS 적용	b1, b2, c1, c2, d2, e2
입력 검증 및 필터링	서버리스 함수, API 요청 등 외부 입력에 대한 유효성 검사 강화	c1, c2, d1, d2
가시성 및 모니터링 강화	로그 수집, 감사 로깅, 이벤트 모니터링을 통한 이상 행위 조기 탐지	a2, b1, b2, c1, d1, e1, f2
공급망 관리	SBOM 관리, 외부 소프트웨어 무결성 검증 및 파이프라인 보안 강화	f1, f2
인프라 구성관리	클라우드 인프라 구성 오류 방지 및 권한 관리 강화	g1, g2

4. 결론

클라우드 네이티브 환경은 컨테이너, 마이크로서비스, 오케스트레이션, 서버리스, 자동화 등 다양한 기술 요소로 구성되어 있으며 이는 높은 민첩성과 유연성을 제공하는 동시에 새로운 보안 위협을 동반한다. 본 논문에서는 [그림 2]와 같이 클라우드 네이티브 환경의 보안 취약성을 주요 기술 위협 유형으로 분류하고, 각 위협에 대한 대응 전략을 제시하였다. 특히, 컨테이너 보안 위협, 오케스트레이션 기반 취약점, 서비스 간 통신 보안, 서버리스 특화 위협, 네트워크 분리 실패, 그리고 공급망 보안 문제는 클라우드 네이티브 환경에서 빈번하게 발생할 수 있으며 이러한 위협을 인식하고 사전 대응하는 것이 중요하다. 본 논문에서 제안한 보안 대응 방안은 이미지 서명 및 취약점 스캐닝, RBAC 기반 최소 권한 설정, 제로 트러스트 네트워크 구성, 실시간 실행 환경 보호, CI/CD 파이프라인 보안 강화 등을 포함한다. 이를 통해 클라우드 네이티브 환경에서의 보안은 통합적이고 전략적인 접근이 필요하다는 점을 강조하였다. 이러한 분석은 실무 보안 관리자와 개발자에게 유의미한 기준을 제공하며 향후에는 이를 기반으로 클라우드 네이티브 특화 보안 점검 기준 설계 등으로 확장될 수 있다. 결론적으로, 클라우드 네이티브 환경의 보안은 선택이 아닌 필수이며, 선제적이고 구조화된 보안 전략 수립이 조직의 안정적인 디지털 전환을 위한 핵심 요소가 될 것이다.

참고문헌

[1] Cloud Native Computing Foundation (CNCF), Cloud Native 2024: 코드, 클라우드, 변화의 10년을 향하여 (Cloud Native 2024: Approaching a Decade of

Code, Cloud, and Change), 2025. Retrieved from https://www.cncf.io/reports/cncf-annual-survey-2024/
[2] 한국지능정보사회진흥원(NIA), 클라우드 네이티브 정보시스템 구축 발주자 안내서, 대구, 한국지능정보사회진흥원, 2022.
[3] Pratyusa K. Manadhata and Jeannette M. Wing, "An Attack Surface Metric," IEEE Transactions on Software Engineering, vol. 37, no. 3, pp. 371 - 386, 2011.
[4] IBM, "Attack Surface Management (ASM)," 2024. Retrieved from https://www.ibm.com/kr-ko/topics/attack-surface-management
[5] National Institute of Standards and Technology (NIST), SP 800-207: Zero Trust Architecture, Gaithersburg, NIST, 2020.
[6] National Institute of Standards and Technology (NIST), SP 800-190: Application Container Security Guide, Gaithersburg, NIST, 2017.
[7] Cloud Native Computing Foundation (CNCF), Cloud Native Security Whitepaper, Cloud Native Computing Foundation, 2020.
[8] OWASP, Kubernetes Top 10 Security Risks, OWASP Foundation, 2022.
[9] 성기수, "STRIDE 위협 모델링에 기반한 클라우드 컴퓨팅의 쿠버네티스(Kubernetes)의 보안 요구사항에 관한 연구," 한국정보보호학회 학술대회논문집, 서울, 2021, pp. 231 - 234.
[10] 김현우, 이지은, "국내 금융권 클라우드 보안 위협 및 보안 요구사항에 관한 연구 (A Study on Cloud Security Threats and Security Requirements in the Domestic Financial Sector)", 한국차세대컴퓨팅학회 논문지, 제20권, 제8호, pp. 77 - 85, 2024.