Zero Trust 아키텍처 구현을 위한 쿠버네티스 워크로드 보안 연구 동향†

한창희*, 이석민*, 신영주**

*, **고려대학교(대학원생, 교수)

A Survey on Kubernetes Workload Security for Zero Trust Architecture

Changhee Han*, Seokmin Lee*, Youngjoo Shin**

*, ** Korea University(Graduate student, Professor)

요 약

최근 IT 서비스 구축을 위해 컨테이너를 활용한 분산 시스템의 사용이 증가하고 있다. 이에 따라 시스템의 보안성 향상을 위한 Zero Trust 보안 패러다임의 필요성이 대두되고 있다. Zero Trust란 시스템 내 암묵적인 신뢰 구간 및 장비를 제거하여, 서비스 자원 접근 시 요청자의 권한을 항상 확인하는 보안 패러다임이다. 그러나 분산 시스템 플랫폼에서 대표적으로 활용되는 쿠버네티스 아키텍처는 기본적으로 Zero Trust를 충족하지 못한다. 이에, 본 연구는 쿠버네티스 환경에서 Zero Trust 아키텍처를 구현하기 위한 기존 연구 동향을 조사하고, 각각의 방법들을 비교 분석하여 그 한계점에 관해 기술한다. 최종적으로, 기존 방법들에서 구현하지 못한 시스템 내의 신뢰 구간 제거의 필요성을 강조한다.

I. 서론

쿠버네티스는 컨테이너로 구성된 분산 시스템을 관리하기 위한 컨테이너 오케스트레이션 플랫폼이다. 이러한 클라우드 관리 플랫폼은 최근 클라우드 서비스의 규모가 커짐에 따라 수요가 증대하고 있다. 대표적인 클라우드 서비스 기업인 GCP, AWS, Azure에서 모두 쿠버네티스를 통한 컨테이너 관리 서비스를 제공한다.

쿠버네티스의 보안 기능은 서비스 사용자에 대한 인증 및 인가를 중심으로 이루어져 있다. 즉, TLS 통신을 사용해 서비스 사용자의 인증서를 확인하고, 사용자에게 최소한의 서비스 접근 권한을 설정하여 필요한 만큼의 서비스 리소스만 사용할 수 있도록 하고 있다. 그러나 쿠버네티스의 워크로드 간 통신에 대해서는 보안 기능을 제공하지 않는다. 사용자가 워크로드에 접근하기 전에 인증 및 인가 과정을 거치기 때문에, 접근한 사용자는 권한이 있는 사용자로 취급한다. 이처럼 특정 보안 조치를 통과한 사용자에게 암묵적인 신뢰를 부여하는 보안 모델을 경계선 보안 모델이라고 한다.

경계선 보안 모델은 시스템 아키텍처의 일반적인 보안 모델이지만, 네트워크 서비스에 대한 다양한 접근이 가능함에 따라 보안 경계선의 신뢰성이 낮아지고 시스템의 침해 위험성이 높아지고 있다. Zero Trust는 이런 위험성을 보완할 수 있는 보안 패러다임이다. 이는 자원 보호에 중점을 둔 보안 패러다임으로, 경계선 내 암묵적인 신뢰를 제거하고 신뢰성을 항상 확인해야 한다는 점을 전제로 한다. 쿠버네티스 또한 경계선 보안 모델을 사용하기에, 쿠버네티스를 사용한 서비스를 Zero Trust 아키텍처로 구현하기 위한 다양한 연구가 진행되었다.

본 논문에서는 쿠버네티스에 Zero Trust를

† 본 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (2019-0-00533, 컴퓨터 프로세스의 구조적 보안 취약점 검증 및 공격 탐지 대응)

적용하는 최신 연구 동향을 조사한다. 그리고 각 연구에서 제시한 Zero Trust 아키텍처의 구조를 분석하고 각각의 한계점을 비교한다.

본 논문의 구성은 다음과 같다. 2절에서는 쿠버네티스 아키텍처와 Zero Trust를 소개한다. 3절은 Zero Trust Artchitecture 구조를 분석한다. 4절에서는 분석한 내용의 비교 결과를 제시한다. 마지막으로 5절에서는 결론을 기술한다.

II. 배경 지식

2.1. 쿠버네티스 아키텍처

쿠버네티스 아키텍처는 그림 1과 같이 하나의 마스터와 1개 이상의 워커 노드로 구성되어 있다. 마스터는 워커 노드에 서비스를 실행하고 시스템 설정 정보를 관리하는, 클러스터 전체에 대한 제어 장치를 담당한다. 그리고 워커 노드는 파드(pod)를 호스팅하여 서비스를 제공한다. 파드는 한 개 이상의 컨테이너로 구성된 쿠버네티스의 서비스를 배포하는 최소 단위로서 쿠버네티스의 워크로드이다.

쿠버네티스는 역할 기반 접근 제어를 기본으로 사용자의 접속을 통제한다. 사용자가 서비스 접속 요청 시, 마스터의 API 서버가 이를 수신하여 사용자 인증 및 권한 확인을 통해 접속 허용 여부를 결정한다. 사용자 인증은 사용자와 API 서버 간 TLS 통신 과정에서 TLS 인증서를 통해 이루어진다.

2.2. Zero Trust

Zero Trust는 이미 침해가 이루어져 아무것도 신뢰할 수 없다는 것을 가정하고 피해를 최소화하는 것을 목표로 하는 새로운 보안 패러다임이다. 시스템을 대상으로 하는 모든 접속 요청에 대한 인증 및 인가 조치를 통해 통신 내용에 대한 로깅 및 모니터링의 필요성을 강조한다.

미국 국립 표준기술 연구소(NIST)에서는 2009년 Zero Trust 아키텍처에 대한 표준을 발표했다 [1]. 해당 표준은 기업에서 서비스 구축 시, Zero Trust 아키텍처 형태로 구축하기 위한 7원칙을 제시한다. 기본 원칙을 제시하는 것과 동시에, 위 원칙을 모두 지키는 것은 쉽지 않지만 가능한 모든 원칙을 지키기 위한 노력의 중요성을 강조한다.

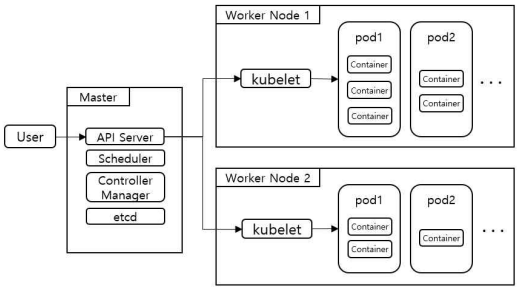

그림 1 쿠버네티스 아키텍처 개요

III. Zero Trust 아키텍처

쿠버네티스의 보안 기능은 API 서버를 경계로 하는 경계선 보안 모델을 사용한다. 그러나 해당 보안 기능은 서비스 사용을 위한 사용자의 접근은 보호하되, 워크로드 간 통신에 대한 보안 조치는 이루어지지 않는다. 이는 “네트워크 위치와 관계없이 모든 통신 사항은 보호되어야 한다”는 NIST의 Zero Trust 아키텍처 구현 원칙을 준수하지 않음을 의미한다. 이에, 본 절에서는 쿠버네티스 기본 보안 기능과 솔루션을 활용하여 워크로드 간 통신을 보호하려는 기존의 기술 및 관련 연구들에 대해 조사한 내용을 기술한다.

3.1. 쿠버네티스 자체 보안 기능

쿠버네티스 자체 기능을 활용해 워크로드 보안 조치를 수행할 수 있다 [2]. 워크로드 실행 권한을 최고 관리자가 아닌 사용자 권한으로 하여, 워크로드가 공격자에게 악용되어도 공격자의 권한을 제한하여 피해를 최소화할 수 있도록 한다. 그리고 워크로드 간 네트워크 정책을 설정하여 IP, 포트 등의 네트워크 요소를 기준으로 각 워크로드에 접근할 수 있는 사용자를 제한할 수 있다.

하지만 이런 자체적인 조치는 Zero Trust 관점에서 충분하지 못하다. 워크로드 실행 시, 관리자가 아닌 사용자 권한으로 실행할 것을 제안하지만, 사용자 권한으로도 네트워크 접속, 코드 실행 등 워크로드의 취약점을 공략하는 행위를 수행할 수 있다. 네트워크 정책을 통한 보안은 네트워크 속성 기반 접근 제어를 사용하지만, 세부적인 제

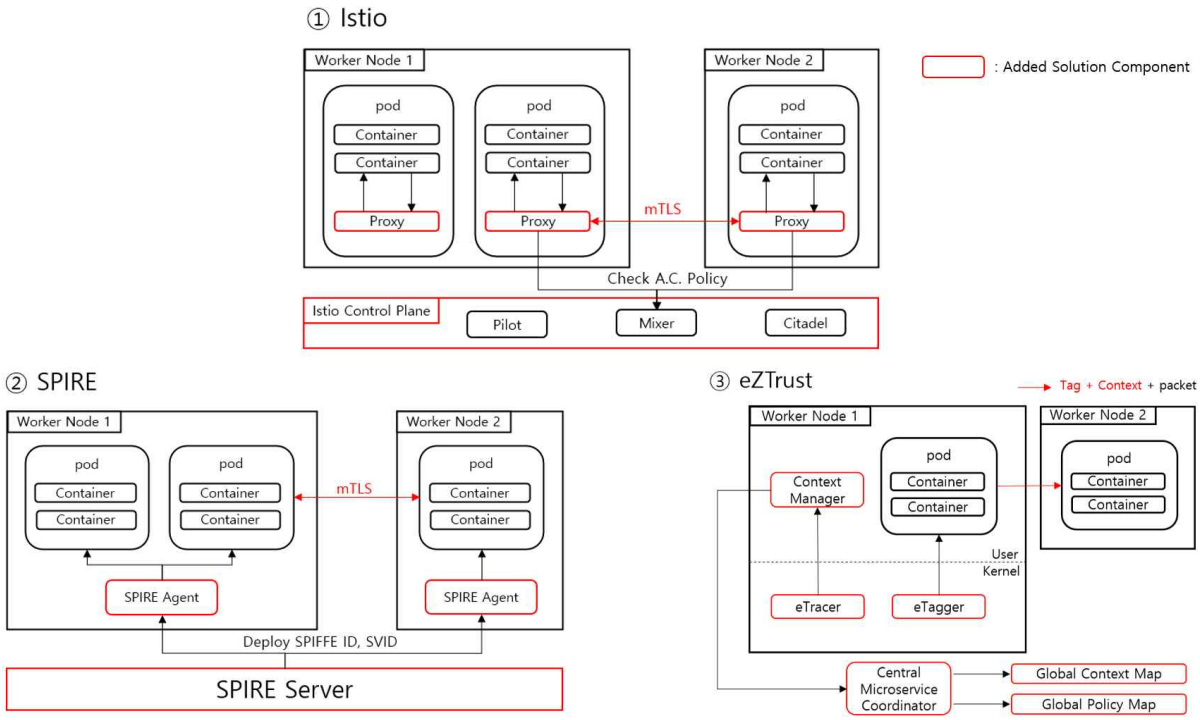

그림 2 쿠버네티스 기반의 Zero Trust 아키텍처

어는 불가능하고 분산 서비스 특성상 워크로드의 네트워크 속성의 변화가 자주 발생되기에 정책 관리를 위한 리소스를 과도하게 필요로 한다.

3.2. Istio

Istio [3]는 워크로드 간 데이터 공유 기반을 제공하는 오픈소스 service mesh 플랫폼이다. 사이드카 디자인 패턴을 사용하여, 그림 2의 ①과 같이 모든 워크로드 간 통신을 사이드카 프록시를 통해 수행한다. 사이드카 프록시는 중앙의 Istio Control Plane를 통해 제어한다.

Istio의 주요 보안 기능은 mTLS(Mutual TLS)를 사용한 통신 암호화이다. 새로운 파드를 생성하는 경우, Istio Control Plane의 Citadel은 생성한 파드를 대상으로 X.509 인증서를 발급한다. 인증서를 발급받은 파드는 시스템 내 다른 파드에 접속 시, 해당 인증서를 사용해 TLS 세션 연결을 요청한다. 접속 요청을 받은 파드는 Istio Control Plane의 Mixer를 통해 요청자에 대한 접근 정책을 확인한 뒤 접속 요청자에게 인증서를 전달하여 세션을 맺는다. 이를 통해 워크로드 간 통신에 대하여 인증 및 인가를 수행하고, 통신 내용을 암호화한다.

3.3. SPIRE

SPIRE [4]는 SPIFFE를 사용하여 구현한 워크로드를 대상으로 하는 인증서 발급 및 배포 시스템이다. SPIFFE 분산 시스템을 안전하게 식별하기 위한 오픈소스 표준으로, X.509 인증서를 기반으로 하는 SVID(SPIFFE Verifiable Identity Document)를 사용하여 워크로드 간 접속자 인증 및 암호화 기능을 제공한다.

SPIRE를 구현한 서비스에서 새로운 워크로드 실행 시, 실행한 워크로드는 프로세스 ID를 기반으로 SPIFFE ID를 발급받는다. 그리고 SPEFFI ID를 기반으로 SVID를 발급받아 워크로드 간 통신 시 mTLS를 사용하여 상호 인증 및 통신 암호화를 수행한다. SPIFFE ID와 SVID는 그림 2의 ②와 같이 SPIRE Server에서 발급하고 SPIRE Agent를 대상으로 배포하며, 각 워커 노드마다 실행하는 SPIRE Agent는 발급받은 SPIFFE ID나 SVID를 워크로드로 전달한다.

3.4. eZTrust

eZTrust [5]는 일반적으로 경계선 보안 모델의 경계 기준을 네트워크 엔드 포인트에서 워

크로드로 세분화하는 솔루션이다. eBPF를 사용한 패킷 필터링을 통해 워크로드에 대한 접근을 통제한다.

eZTrust 솔루션을 사용하는 환경에서 각각의 워크로드는 고유한 Tag를 가지며 이에 매칭되는 Context의 조합을 갖는다. 워크로드가 Tag 및 Context를 취득하는 구조는 그림 2의 ③과 같다. 커널에서 실행하는 eTagger는 워크로드 ID와 프로세스 ID를 사용해 워크로드에 할당할 Tag를 생성한다. 그리고 eTracer를 통해 커널에서 발생할 이벤트를 추적하여 Context를 생성하고 Context Manager에 전달한다. 이렇게 전달한 Context는 해당하는 워크로드와 중앙 제어 장치로 전달하여 워크로드 간 통신에 사용된다.

특정 워크로드에서 접속 요청 수신 시, 패킷 내 Tag 값을 통해 접속자의 Context 정보를 취득한다. 취득한 Context 정보는 정책 정보와 대조하여 접속자의 접속 요청을 대상으로 한 접속 허용 여부를 결정한다.

IV. 분석

본 절에서는, 3절에서 기술한 방법들이 Zero Trust 아키텍처의 기준에 적합한지 분석한다. 적합성 판단 기준은 NIST에서 제시하는, 네트워크 관점에서의 Zero Trust 권장 보안 설정 사항을 따른다.

3절에서 기술한 방법들은, 쿠버네티스에 추가적인 솔루션을 활용해 워크로드 보안 기능을 적용하여 Zero Trust를 구현하고 있다. 그러나 분석결과, 조사한 모든 아키텍처에서 Zero Trust를 완벽하게 구현하지는 못하는 것으로 나타났다. 표 2와 같이, 세 아키텍처는 Zero Trust에서 중요시하는 접속자 인증 및 권한 확인, 그리고 워크로드 간 통신 암호화 및 모니터링 기능에 집중하고 있다. 그러나 세 아키텍처 모두 암묵적 신뢰 요소를 완전히 제거하지는 못했다. 해당 아키텍처들은 워크로드 보안을 위한 중앙 제어 장치를 갖게 되는데, 중앙 제어 장치는 신뢰할 수 있는 장치로 간주하여 별도의 보안 조치를 하지 않는다.

이는, 시스템 내 암묵적인 신뢰 제거를 목표로 하는 Zero Trust의 핵심 요건에 반하는 것으로, 완벽한 Zero Trust를 구현하기 위해서는 이에 대한 조치가 필요하다.

	k8s	Istio	SPIRE	eZTrust
워크로드 간 통신 암호화	X	O	O	X
워크로드 간 통신 모니터링	X	O	X	X
접속자 권한 확인	O	O	X	O
접속자 인증	X	O	O	X
암묵적 신뢰 요소 제거 여부	X	X	X	X

표 1 아키텍처 별 워크로드 보안 기능 비교

V. 결론

본 논문에서는 쿠버네티스 아키텍처에 추가 솔루션을 적용하여 Zero Trust 아키텍처를 구현하는 방법들을 조사하였다. 조사한 방법들을 비교 분석한 결과, 단일 솔루션으로는 Zero Trust를 완벽하게 구현하지 못하였으며 암묵적 신뢰 구간을 완전히 제거하지 못하는 것으로 나타났다. 이에, 시스템 내 신뢰 구간을 제거하여 쿠버네티스를 최대한 Zero Trust에 가까운 형태로 구현하는 방법들을 지속적으로 연구할 필요가 있다.

[참고문헌]

[1] S.Rose, O.Borchert, S.Mitchell, S.Connelly, “NIST Special publication 800-207 Zero Trust Architecture”, National Institute of Standards and Technology, 2020

[2] G.Budigiri, C.Baumann, J.T.Mühlberg, E.Truyen and W.Joosen, “Network Policies in Kubernetes: Performance Evaluation and Security Analysis” in Joint European Conference on Networks and Communications & 6G Summit, 2021

[3] C.Weever, M.Andreou, “Zero Trust Network Security Model in containerized environments”, University of Amsterdam, 2020

[4] A.Goel and B.Thangaraju, “Authenticating Distributed System Using SPIRE over Kubernetes Cluster”, in IEEE International Conference on Electronics, Computing and Communication Technologies(CONECCT), 2022

[5] Z.Zaheer, H.Chang, S.Mukherjee and J.Merwe, “eZTrust: Network-Independent Zero-Trust Perimeterization for Microservices”, ACM Symposium on SDN Research, 2019