컨테이너 네트워크 인터페이스의 구조적 분석을 통한 보안 취약점 연구

고형욱° 정경헌 권태경

서울대학교 컴퓨터공학과

kohowo1999@snu.ac.kr tiop7@snu.ac.kr tkkwon@snu.ac.kr

Researching Security Vulnerabilities Through Structural Analysis of Container Network Interfaces

HyeongUk Ko° Gyeongheon Jeong Taekyoung Kwon

Seoul National Univ. CSE

요 약

본 논문은 현대 클라우드 환경의 핵심 요소인 컨테이너 네트워크 인터페이스(CNI)에 대하여, 널리 사용되는 2 가지 CNI 플러그인에서 발생할 수 있는 보안 취약점을 조사하고 그 대응책을 제시하고자 한다. Flannel 에서는 모니터링 도구나 LLM 기반 네트워크 정책 생성 도구를 도입하여 성능 저하 없이 보안성을 강화할 수 있다. Calico 에서는 BGPsec 을 도입하여 취약점에 가해지는 공격을 완화할 수 있다. 본 논문은 CNI 플러그인별 네트워크 보안 대책을 제시함으로써 현 컨테이너 네트워크의 보안을 강화하는 설계 방안을 제안한다.

1. 서론

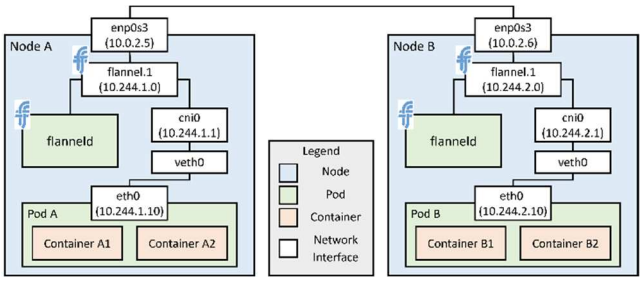
현대 클라우드 네이티브 환경의 핵심 요소인 컨테이너 기반 가상화 기술은 애플리케이션과 그 실행에 필요한 모든 라이브러리, 의존성을 하나의 이미지로 패키징하여 격리된 상태로 실행하는 운영체제 수준의 가상화 기술이다. 이러한 구조는 오버헤드가 적어 배포가 빠르고 이식성이 뛰어나다는 장점이 있어 현대의 DevOps 및 마이크로서비스 아키텍처(MSA)의 표준으로 자리 잡았다.

그러나 해당 기술이 적용된 컨테이너 네트워크는 동적으로 할당되는 IP 주소와 수시로 변하는 토폴로지로 인해 전통적인 방화벽이나 침입 방지 시스템(IPS)으로는 트래픽 제어가 어렵다. 특히 쿠버네티스(Kubernetes)와 같은 오케스트레이션 도구는 컨테이너 네트워크 인터페이스(CNI)를 통해 컨테이너 간 통신을 표준화하는데, 이때 각 CNI 플러그인은 오버레이 네트워크 구성 방식이나 패킷 처리 메커니즘이 상이하며, 이에 따른 고유한 취약점을 내포하고 있다. 따라서 본 연구에서는 두 가지 보편적인 CNI 플러그인의 구조적 취약점을 분석하고, 안전한 네트워크 환경 구축을 위한 대응 방안을 모색한다.

2. 본론

본 논문에서는 현재 널리 쓰이고 있는 CNI 플러그인인 Flannel, Calico 의 보안 취약점을 조사하고 이를 완화할 수 있는 대응책을 제시하였다.

2.1 Flannel



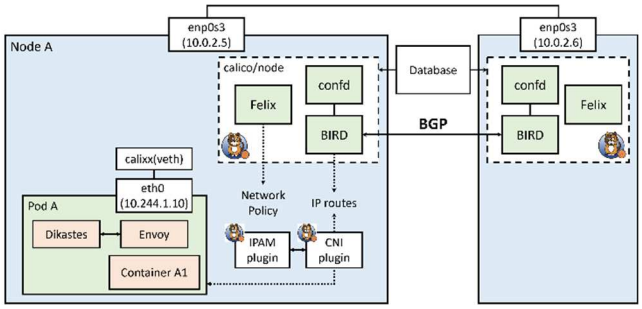
(그림 1) Flannel 이 적용된 컨테이너 네트워크 구성도

Flannel 은 가장 단순하고 설정이 쉬운 CNI 로, 각 노드에 고유한 서브넷을 할당하고 VXLAN(Virtual eXtensible LAN)을 사용하여 패킷을 캡슐화한다. 아키텍처의 핵심인 각 노드의 flanneld 데몬은 클러스터 저장소에 저장된 네트워크 구성을 바탕으로 로

컬 라우팅 테이블을 설정하며, 각 노드에 고유한 서브넷이 할당된 브리지 인터페이스(cni0)를 설치하고 노드 간 트래픽을 VXLAN 인터페이스(flannel.1)로 캡슐화하여 전송한다.[1]

이러한 구조는 서브넷 수준의 격리를 통해 네트워크 경합을 방지하고 클러스터 전반의 IP 주소 관리를 단순화하는 이점을 제공하지만 , 단순성에 초점을 맞춘 설계로 인해 보안 측면에서는 상당한 한계를 지닌다. Flannel 은 자체적인 네트워크 정책을 지원하지 않는데, 이로 인해 트래픽의 이동에 제한이 걸리지 않기 때문에 공격자는 악성 컨테이너를 만들어 내부에 침투한 후, 이를 이용하여 민감한 데이터를 빼돌리거나 다른 컨테이너에 서비스 거부 공격을 할 수 있다. 이러한 공격은 노드별로 모니터링 데몬을 하나 더 추가하거나, flanneld 자체에 모니터링 기능을 추가하여 악성 트래픽을 감지하는 방식을 통해 완화할 수 있다. 또한 이를 통해 Flannel 의 자체 성능을 크게 낮추지 않으면서 그 보안성을 향상시킬 수 있다. 추가로 LLM 을 이용하여 관리자가 요구하는, 또는 이전에 감지했던 악성 트래픽의 전송을 막는 네트워크 정책을 생성하는 도구를 적용하는 방안도 고려해볼 수 있다.

2.2 Calico



(그림 2) Calico 가 적용된 컨테이너 네트워크 구성도

Calico 는 대규모 클러스터 환경에 적합하도록 설계된 CNI 로, BGP 기반 라우팅을 채택하여 효율적인 노드 간 통신을 지원한다. 아키텍처의 핵심은 각 노드에 배포되는 Calico 에이전트(calico/node)이며, 이는 3 가지 컴포넌트로 구성된다. Felix 는 파드별 라우팅 설정과 iptables 규칙 기반의 트래픽 제어를 담당하고, BIRD 는 노드 간 라우팅 정보를 교환하며, confd 는 데이터베이스의 정보를 기반으로 노드 전체의 네트워크 설정을 적용한다. 이외에 IPAM 플러그인은 노드 내부의 IP 주소 할당 업무를 전담한다.[2]

보안적 측면에서 Calico 는 기본 Kubernetes 정책 구조를 확장하여 네임스페이스 및 클러스터 단위의 커스텀 정책을 제공하며, 정책 적용 시 우선순위 지정과 더불어 Allow, Deny, Log, Pass 등 세분화된 동작을 명시할 수 있어 계층적이고 유연한 트래픽 제어가 가능하다. 또한 Flow Logs 기능을 통해 트래픽 분석을 지원하는데, 이는 Splunk 와 같은 외부 도구와 통합되어 빠른 이상 탐지를 가능케 한다. 다만 Calico 는 결국 BGP 를 이용한다는 점 때문에 해당 프로토콜이 가지고 있는 취약점을 그대로 답습한다. 이를 이용하여 공격자는 악성 컨테이너를 만들어 내부 네트워크에 침투한 후, BGP 하이재킹 공격을 통하여 트래픽을 탈취하는 등 중간자 공격을 수행할 수 있다. 따라서 이를 완화하기 위해 BGP Peer 에 비밀번호를 설정하거나 전체 경로를 검증하여 경로 조작을 막는 BGPsec[3]을 Calico 아키텍처에 적용하는 방안을 고려해볼 수 있다.

3. 결론

본 논문에서는 현대 클라우드 환경에서 널리 쓰이는 도구인 CNI 플러그인의 구조적 특성을 분석하고, 해당 구조에서 발생할 수 있는 보안 취약점을 분석하여 그 대안을 제시하였다.

Flannel 은 자체적인 네트워크 정책을 지원하지 않기 때문에 악성 트래픽을 감지할 수 있는 모니터링 도구와 트래픽을 제어할 수 있는 정책을 생성하는 방안이 중요했고, Calico 는 BGP 를 사용하기 때문에 해당 프로토콜에 같이 따라오는 취약점을 완화하기 위해 BGP Peer 에 비밀번호를 설정하거나 BGPsec 을 도입하는 것이 중요했다. 본 논문에서는 이와 같은 보안 대책을 제시함으로써 현 컨테이너 네트워크의 보안을 강화하는 설계 방안을 제안하였다. 추후 연구에서는 제안된 보안 대책을 실제로 적용해본 후 보안성이 얼마나 강화되었는지 정량적으로 분석해보고, Cilium, Weavenet 등 다른 CNI 플러그인에 대해서도 분석하여 보다 풍부한 환경에서의 보안 설계를 연구하고자 한다.

4. Acknowlegement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2026-2021-0-02048)

5. 참고 문헌

[1] B. Kim, J. Kim and S. Lee, “Exploring Security Enhancements in Kubernetes CNI: A Deep Dive Into Network Policies”, *IEEE Access*, vol. 13, pp. 35322-35338, Feb. 2025.

[2] Tigera, “Component architecture”. (https://docs.tigera.io/calico/latest/reference/architecture/overview)

[3] M. Lepinski and K. Sriram, “BGPsec Protocol Specification”, Sep. 2017. (https://datatracker.ietf.org/doc/html/rfc8205)