eBPF 기반 실시간 계층적 Visibility 를 이용한 클라우드-네이티브 엣지 클러스터에서의 보안 훈련

김민석, 김종원*

광주과학기술원 전기전자컴퓨터공학부, 광주과학기술원 AI 대학원*
wsms8646@gist.ac.kr, jongwon@gist.ac.kr

Secure Training over Cloud-Native Edge Cluster employing eBPF-based Real-time hierarchical Visibility

Minseok Kim, Jong Won Kim
GIST (Gwangju Institute of Science & Technology)

요 약

본 논문은 클라우드-네이티브 엣지 컴퓨팅을 이용하는 학습자들의 위험한 행동을 차단하면서 정해진 단계별 학습 코스를 효과적으로 학습하도록 설계했던 Playground 교육환경에서 일어날 수 있는 SPOF(Single Point Of Failure)를 해결하기 위해 계층화를 도입하는 설계를 제안한다.

Ⅰ. 서 론

클라우드 네이티브 엣지 컴퓨팅은 더 다양하고 time-sensitive 한 니즈에 효율적으로 대응하기 위해 시스템에 존재하는 IT 리소스들을 추상화하고 필요한 곳에 적재적소의 리소스들을 스케줄링하는 방식이다. 그렇기에 클라우드-네이티브 엣지 클러스터를 통한 구현으로 각 학습자들에게 적합한(self-fitted) 환경을 빠르게 제공하는 교육환경이 각광받고 있다. 하지만 하나의 클러스터를 여러 테넌트가 함께 이용하는 멀티-테넌트 환경에서는 일부 무지한 테넌트의 행동으로 인해 다른 테넌트와 전체 시스템에 악영향을 줄 수 있기 때문에 이를 방지하기 위한 방지책이 중요하다.

멀티-테넌트인 교육환경이기에서는 무지한 테넌트로 인한 문제는 언제든지 일어날 수 있기에 본 논문에서는 기존에 보안 관련 관심사였던 외부에서의 공격보다는 무지한 테넌트로 인한 내부 보안 문제를 다루려 한다. 다만 추상화를 통한 클라우드-네이티브 엣지 컴퓨팅 환경은 기존의 모노리스 방식에 비해 구조가 복잡해졌기에 내부 보안 문제를 다루기 위한 실시간 visibility 를 효과적으로 구현하기 위해서는 최대한 오버헤드를 줄일 필요가 있다. 그렇기에 커널에서 사용자 소스코드를 실행하여 불필요한 context-switching 오버헤드를 줄일 수 있는 eBPF(extended Berkerley Packet Filter)를 사용하여 구현한 Playground 교육환경을 개선하려 한다[2].

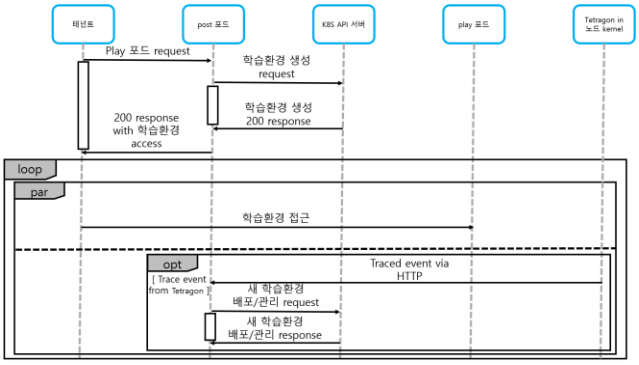

그림 1. Playground 교육환경에 대한 sequence chart

그림 1 은 기존의 Playground 교육환경에서 어떻게 각 테넌트들의 학습환경을 배포/관리하는지에 대한 sequence chart 이다[1]. sequence chart 를 보면 모든 테넌트와 play 포드들은 모두 하나의 post 포드에게 학습환경 배포/관리를 맡기기 때문에 하나의 post 포드에게 많은 권한과 로드가 부여됨을 알 수 있다. 그렇기에 본 논문에서는 post 포드를 계층화하여 과도한 권한 및 로드를 여러 컴포넌트에게 분배하려 한다.

Ⅱ. Playground 교육환경 관련 용어

Playground 교육환경을 개선하기 전에 Playground 교육환경의 구성요소들을 정리하려 한다. Playground 교육환경의 구성요소는 크게 학습자라고 할 수 있는 테넌트, 테넌트의 실제 학습 공간인 play 포드와 테넌트의 요구사항에 맞는 play(활용) 포드를 배포/관리하는 post(초소) 포드가 있다. 그리고 post 포드가 play 포드를 배포/관리하기 위한 기능을 집행해주는 Kubernetes API 서버와 클러스터 내의 각 노드의 kernel 에 존재하며 play 포드가 하는 행동을 감시하고 post 포드에게 알려주는 eBPF 를 이용한 구현체인 Tetragon 이 있다. 그림 1 의 sequence chart 에서 보이듯이 테넌트에서 post 포드에게 play 포드 request 를 하면 Kubernetes API 서버를 통해서 생성한 play 포드를 테넌트가 사용한다. 이후에도 post 포드는 각 노드의 kernel 에 있는 Tetragon 에게서 play 포드에 대해 감지한 이벤트를 보고 받고 상황에 따라 Kubernetes API 서버를 통해 새로운 play 포드를 배포한다. 이 과정을 보면 각 play 포드는 play 포드를 사용하는 테넌트만 담당하고 노드 kernel 에 있는 Tetragon 도 노드에 있는 play 포드들만 담당하지만 post 포드는 클러스터에 있는 모든 play 포드를 관리해야 하기 때문에 개선된 Playground 교육환경에서는 post 포드를 계층화하여 각 post 포드가 담당하는 영역을 줄이고자 한다.

Ⅲ. 계층화의 개념과 필요성

계층화는 하나의 큰 component 를 여러 개의 계층인 sub-component 로 나누어 component 의 기능을 나누어

가지는 것을 의미한다. 3 에서 제시한 Playground 환경은 하나의 초소 포드에 너무 많은 권한과 로드를 주었기에 초소 포드의 오작동이 Playground 교육환경 전체에 대한 중단을 초래하는 SPOF(Single Point Of Failure)의 문제를 발생시킬 수 있다. 그렇기에 본 논문에서는 계층화를 통해 기존 Playground 환경 속 초소 포드가 가진 권한과 로드를 여러 sub-component 에 분산시켜 더욱 안정적인 Playground 교육환경을 제시하려 한다.

Ⅳ. 영역별 초소 포드를 이용한 layering Playground

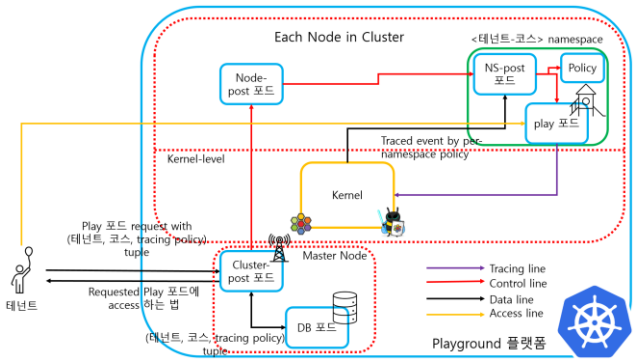

그림 2. Layering Playground 교육환경의 배포/관리

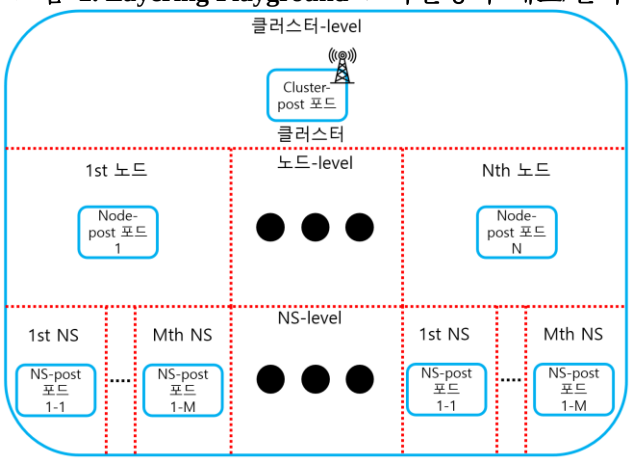

그림 3. Layering 계층화의 영역별 관리 구조

layering 계층화는 영역별 초소 포드를 이용한 계층화이다. 여기서 영역이란 Kubernetes 에서 일반적으로 사용하는 단위들을 표현한 것이며 layering 계층화에서는 클러스터, 노드, 학습환경(Namespace) 3 가지의 영역을 사용할 것이다. layering 을 통한 계층화는 최상위계층 클러스터부터 최하위계층인 학습환경까지 클러스터, 노드, 학습환경 순으로 영역을 나누는 계층화이다. layering Playground 환경에서는 각 학습환경의 활용 포드와 추적 정책을 그 학습환경에 있는 NS-초소 포드가 관리하고 특정 노드에 존재하는 NS-초소 포드들을 그 노드의 Node-초소 포드가 관리한다. 그리고 마지막으로 클러스터에 존재하는 모든 Node-초소 포드들을 Cluster-초소 포드가 관리한다. 즉, 각 영역에 존재하는 초소 포드들이 하위 영역의 초소 포드들이나 활용 포드들을 관리하여 기존 초소 포드의 권한과 로드를 여러 초소 포드들에게 분산시키는 계층화다. layering Playground 환경으로 바뀌면서 기존과 달라진 점은 크게 2 가지 장점과 1 가지 단점이 있다. 첫 번째 장점은 특정 초소 포드가 문제가 생겨서 자신의 역할을 다하지 못하더라도 클러스터 전체에 미치는 영향이 매우 적다. NS-초소 포드가 문제가 생기면 NS-초소 포드가 관리하는 학습환경은 재배포/관리를 하기 힘들어지지만 그 이외의 영역은 아무런 문제 없이 추적 정책을 통해 tracing 하고 문제가 생기면 재배포/관리를 할 수 있다. 설령 Cluster-초소 포드에 문제가 생긴다고 하더라도 하위 초소 포드들에게 문제가 없다면 각 Node-초소 포드들은 NS-초소 포드들을 관리하고 NS-초소 포드들이 활용 포드들을 관리하며 학습환경의 관리에 큰 문제가 없을 것이다. 두 번째 장점은 로드의 분산이다. 기존의 Playground 환경은 모든 tracing 의 결과가 하나 뿐인 초소 포드로 보내지고 그 초소 포드에서 재배포/관리를 생각해야 했기에 하나의 초소 포드가 클러스터 전체의 로드를 감당해야 했으며 이로 인해 학습환경(namespace)들이 많아지면 많아질수록 그에 비례해 하나의 포드에 걸리는 로드가 커졌다. 하지만 layering Playground 환경에서는 각 학습환경의 tracing 결과는 각 학습환경을 관리하는 NS-초소 포드로 보내지고 NS-초소 포드에서 처리되기 때문에 layering Playground 환경을 확장해도 한 포드에 가해지는 로드로 인한 불안정함은 거의 생기지 않을 것이다. 하지만 이 두 가지 장점을 얻기 위해 생긴 단점이 하나 있다. 기존 Playground 환경에서는 초소 포드에서 바로 학습환경을 만들었지만 layering Playground 환경에서는 Cluster-초소 포드, Node-초소 포드, NS-초소 포드를 거쳐야 비로소 학습환경을 만들 수 있기 때문에 기존의 환경보다 새로운 학습 환경 초기화에 시간이 오래 걸린다는 것이다. layering Playground 교육환경은 초기화가 오래 걸린다는 단점이 있지만 계층화를 통해 안정성과 확장성을 갖춘 환경이 되었다고 할 수 있다.

Ⅴ. 검증 결과 및 결론

본 논문에서는 기존 Playground 교육환경의 취약점이었던 단일 포드에게 과한 권한과 로드가 걸리는 SPOF(Single Point Of Failure) 해결을 위한 layering 계층화를 구상하였다.

layering 계층화는 권한과 로드를 각 영역을 담당하는 초소 포드에게 분산시키며 SPOF 해결에 도움을 줄 수 있으나 새로운 학습환경의 초기화에는 더 오랜 시간이 걸린다는 단점이 생겼다. 그러나 이 단점은 thread pool 기법을 활용한 학습환경 pool 을 통한 학습환경 초기화 단축을 통해 극복할 수 있을 것이기에 이후 layering Playground 환경의 안정성, 확장성을 검증하는 테스트를 진행하며 함께 구현할 예정이다.

감사의 글

이 논문은 2023 년도 정부(과학기술정보통신부)의 재원으로 정보통신산업진흥원의 지원(No. S0101-23-1002, 악천후 등 외부환경 대응 가능한 V2X 기반 connected 플랫폼 기술 개발)과 2023 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No. 2019-0-01842, 인공지능대학원지원(광주과학기술원))을 받아수행된 연구임.

참 고 문 헌

[1] Minseok Kim et al.," eBPF 기반 실시간 Visibility 를 이용한 클라우드-네이티브 엣지 클러스터에서의 보안 훈련" *KICS Summer Conference 2023*, June 21,2023.

[2] Jed Salazar et al.,*Security Observability with eBPF*, O'Reilly Media, Inc., (pp. 23-42), April 05,2022.