[붙임5] 2026년 소・중・대 산학협력프로젝트(캡스톤디자인) 결과보고서

2026년 전남대학교 소프트웨어중심대학사업
소·중·대 산학협력프로젝트(캡스톤디자인) 결과보고서

프로젝트명	Kubernetes 공격 전파 흐름 탐지·시각화 시스템					
Github url 주소	https://github.com/yew0n12/capstone-poding.git					
팀 명	푸딩			과제수행기간	2026. 3. 25. ~ 6. 23.	
지도교수	학 과	인공지능학부		성 명	박태준	
프로젝트 수행인원 (※팀장은 첫줄에 기입)	학과(부·복수전공)	학년	학번	성명	연락처	E-Mail
	인공지능학부	4	235272	나예원	010-5017-3668	wye25503@gmail.com
	인공지능학부	4	235786	김태림	010-4797-9376	ktr0706@naver.com
	인공지능학부	4	230703	양시은	010-4097-7240	didtldms2525@naver.com
	인공지능학부	4	222162	조예지	010-2534-6447	xq1533yy@naver.com
참여 기업	기업명	멘토명		직위	연락처(HP)	E-Mail
	㈜티제이에코시스	황민		이사	010-5613-1142	dorim62@hanmail.net

위와 같이 2026년 전남대학교 소프트웨어중심대학사업
산학협력프로젝트(캡스톤디자인) 지원 프로그램 결과보고서를 제출합니다.

2026년 6월 22일

신청자명(대표학생) : 나예원 (인)
지도교수 : 박태준 (인)

전남대학교 소프트웨어중심대학사업단장 귀하

산학협력프로젝트(캡스톤디자인) 결과보고서(요약)

프로젝트명	Kubernetes 공격 전파 흐름 탐지·시각화 시스템		
수행기간	2026. 3. 25. ~ 6. 23.	소요예산	476,742원
소요예산 세부내역	-회의비: 385,500원 -SW활용비: 91,242원 -재료비: 0원		

참여인원	구분	인원수	성명(모두 기재)
	교수	1	박태준
	석박사과정	-	
	학부생	4	나예원, 김태림, 양시은, 조예지
	기업체	1	황민
	계	6	

추진배경

Kubernetes는 여러 애플리케이션을 컨테이너 기반으로 배포·운영할 수 있어 클라우드 네이티브 환경에서 널리 활용되고 있다. 그러나 컨테이너가 호스트 노드의 커널을 공유하고, 클러스터 내부의 Pod 간 통신이 비교적 자유롭게 구성될 수 있다는 특성으로 인해 하나의 Pod가 침해될 경우 다른 Pod 또는 클러스터 내부 자원으로 공격이 확산될 가능성이 존재한다. 기존 보안 도구는 시스템콜 또는 네트워크 로그 중 하나의 관점에 집중하는 경우가 많아, 공격이 Pod 내부에서 어떻게 진행되고 다른 Pod로 어떻게 전파되는지 전체 흐름을 파악하는 데 한계가 있다. 이에 본 프로젝트에서는 Kubernetes 환경에서 발생하는 이기종 보안 이벤트를 통합적으로 분석하여 공격의 진행 단계와 전파 경로를 추적하는 보안 솔루션을 개발하고자 하였다.

목표 및 내용

본 프로젝트의 목표는 Kubernetes 컨테이너 환경에서 Pod 내부 악성 행위와 Pod 간 외부 전파 흐름을 함께 분석하는 이기종 상관분석 기반 보안 솔루션 "Pod-ing"을 구현하는 것이다. 이를 위해 Falco를 활용하여 Pod 내부 시스템콜 기반 이벤트를 수집하고, Cilium Hubble을 활용하여 Pod 간 네트워크 통신 이벤트를 수집하였다. 수집된 서로 다른 형식의 로그는 공통 심볼 형태로 변환한 뒤 NFA 기반 상태 전이 모델을 통해 각 Pod의 공격 진행 상태를 추적하였다. 또한 Pod 간 통신 이벤트를 공격 흐름의 일부로 포함하여 특정 Pod에서 시작된 공격이 다른 Pod로 확산되는 과정을 하나의 공격 체인으로 분석할 수 있도록 하였다.

시스템은 크게 이벤트 수집부, 로그 정규화 및 상관분석 엔진, 결과 내보내기 모듈, 시각화 대시보드로 구성된다. 분석 결과는 Prometheus와 Grafana를 통해 시각화되며, 사용자는 각 Pod의 상태와 공격 전파 관계를 대시보드에서 확인할 수 있다. 최종적으로 poding-detector, exporter, Grafana dashboard를 포함한 Helm Chart 패키지를 구성하여 다른 Kubernetes 환경에서도 설치 및 활용할 수 있도록 하였다.

기대효과

본 프로젝트를 통해 Kubernetes 환경에서 단일 로그 기반 탐지의 한계를 보완하고, 시스템콜과 네트워크 이벤트를 함께 고려한 공격 흐름 기반 탐지가 가능해졌다. Pod-ing은 단순히 개별 이벤트의 이상 여부를 판단하는 것을 넘어 공격이 어떤 순서로 진행되었고 어느 Pod까지 확산되었는지를 추적할 수 있다. 이를 통해 보안 담당자는 침해 범위와 공격 경로를 빠르게 파악할 수 있으며, 대응 우선순위 결정과 사고 분석에 활용할 수 있다. 또한 Helm Chart 기반 배포 구조를 제공함으로써 향후 다양한 Kubernetes 클러스터 환경에 적용 가능한 오픈소스형 보안 도구로 확장될 수 있다.

1. 프로젝트 개요

프로젝트명	**Kubernetes 공격 전파 흐름 탐지·시각화 시스템**
주제영역	□ 생활 ■ 업무 □ 공공/교통 □ 금융/핀테크 □ 의료 □ 교육 □ 유통/쇼핑 □ 엔터테인먼트
기술분야	□ IoT □ 모바일 □ 데스트톱 SW □ 인공지능 ■ 보안 □ 가상현실 □ 빅데이터 □ 자동제어기술 □ 블록체인 □ 영상처리 □ 기타( )
성과목표	■ 논문게재 및 포스터발표 □ 앱등록 □ 프로그램등록 □ 특허 □ 기술이전 □ 실용화 □ 공모전(*공모전명*) □ 기타( )

2. 프로젝트 추진배경

최근 클라우드 네이티브 환경의 확산으로 Kubernetes는 컨테이너 기반 애플리케이션을 배포하고 운영하기 위한 핵심 플랫폼으로 자리 잡고 있다. Kubernetes는 여러 서비스와 애플리케이션을 Pod 단위로 관리하며, 서비스 간 통신, 자동 확장, 장애 복구, 배포 자동화 측면에서 높은 효율성을 제공한다. 이러한 특성으로 인해 기업과 기관에서는 대규모 서비스를 보다 유연하게 운영하기 위해 Kubernetes 기반 인프라를 적극적으로 도입하고 있다.

그러나 Kubernetes의 편리한 운영 구조는 보안 측면에서 새로운 위험 요소로 작용할 수 있다. Kubernetes 환경에서는 여러 Pod가 하나의 클러스터 안에서 함께 동작하고, 서비스 운영을 위해 Pod 간 통신이 비교적 자유롭게 허용된다. 또한 같은 노드에서 실행되는 Pod들은 커널과 네트워크 자원을 공유하는 특성을 가진다. 이러한 구조는 정상적인 서비스 운영에는 효율적이지만, 공격자의 관점에서는 하나의 Pod를 장악한 뒤 다른 Pod나 클러스터 내부 자원으로 이동할 수 있는 공격 경로가 될 수 있다.

특히 Kubernetes 환경에서의 공격은 단일 이벤트로 끝나는 경우보다 여러 단계에 걸쳐 진행되는 경우가 많다. 공격자는 먼저 취약한 Pod에 침투한 뒤, 해당 Pod 내부에서 shell 실행, 파일 접근, 환경 변수 확인, ServiceAccount token 접근, Kubernetes API 접근 등의 행위를 통해 정보를 수집할 수 있다. 이후 확보한 정보나 권한을 바탕으로 인접한 Pod와 통신하거나 다른 네임스페이스 또는 클러스터 내부 자원으로 이동을 시도할 수 있다. 따라서 단순히 최초 침입 지점을 탐지하는 것만으로는 실제 공격이 어떤 순서로 진행되었고, 어느 Pod까지 영향을 미쳤는지 파악하기 어렵다.

기존의 보안 탐지 방식은 주로 시스템콜 로그 또는 네트워크 로그 중 하나의 관점에 집중하는 경우가 많다. 시스템콜 기반 탐지는 Pod 내부에서 발생하는 프로세스 실행, 파일 접근, 권한 관련 행위 등을 세밀하게 관찰할 수 있다는 장점이 있다. 그러나 해당 행위가 이후 어떤 Pod와의 연결로 이어졌는지, 공격이 클러스터 내부에서 어떻게 확산되었는지를 설명하기에는 한계가 있다. 반대로 네트워크 기반 탐지는 Pod 간 통신 관계와 네트워크 흐름을 확인할 수 있지만, 해당 통신이 정상 서비스 호출인지, 공격자가 내부 정찰이나 전파를 위해 발생시킨 통신인지 판단하기 어렵다. 즉, 하나의 로그만으로는 공격의 원인과 결과, 내부 행위와 외부 전파의 관계를 종합적으로 해석하기 어렵다.

이러한 한계를 해결하기 위해서는 Kubernetes 환경에서 발생하는 서로 다른 종류의 보안 이벤트를 함께 분석하는 상관분석 방식이 필요하다. Pod 내부에서 발생한 시스템콜 이벤트와 Pod 간 네트워크 통신 이벤트를 연결하여 분석하면, 공격자가 내부에서 어떤 행위를 수행했는지뿐만 아니라 그 행위가 이후 어떤 Pod로 확산되었는지도 함께 파악할 수 있다. 또한 단순히 두 Pod가 통신했다는 이유만으로 공격으로 판단하지 않고, 출발지 Pod의 위험 상태, 신규 연결 여부, 도착지 Pod의 후속 의심 행위를 함께 고려함으로써 오탐을 줄이고 탐지 신뢰도를 높일 수 있다.

이에 본 프로젝트에서는 Kubernetes 컨테이너 환경을 대상으로 시스템콜 이벤트와 네트워크 이벤트를 통합적으로 분석하고 시각화를 제공하는 보안 솔루션 "Pod-ing"을 개발하였다. Pod-ing은 Falco를 통해 Pod 내부 시스템콜 기반 이벤트를 수집하고, Cilium Hubble을 통해 Pod 간 네트워크 통신 흐름을 수집한다. 이후 서로 다른 형식의 로그를 공통 심볼로 변환하고, NFA 기반 상태 전이 모델을 적용하여 Pod 내부 공격 단계와 Pod 간 전파 경로를 함께 추적한다.

본 주제를 선정한 이유는 Kubernetes 환경에서의 보안 위협이 단일 Pod 내부의 이상 행위에 그치지 않고, 클러스터 내부의 여러 Pod로 확산될 수 있다는 점에 주목했기 때문이다. 따라서 본 프로젝트는 기존 단일 로그 기반 탐지의 한계를 보완하고, 공격의 발생 여부뿐 아니라 공격의 진행 과정과 확산 범위를 함께 확인할 수 있는 보안 분석 체계를 구현하는 것을 목적으로 한다. 이를 통해 Kubernetes 기반 서비스 운영 환경에서 보안 담당자가 침해 사고의 원인, 전파 경로, 영향 범위를 보다 빠르고 직관적으로 파악할 수 있도록 지원하고자 하였다.

3. 프로젝트(주제) 목표 및 내용

본 프로젝트의 최종 목표는 Kubernetes 환경에서 Pod 내부 악성 행위와 Pod 간 공격 전파를 동시에 추적할 수 있는 이기종 상관분석 기반 보안 시스템을 구현하는 것이다. 이를 통해 단순한 이벤트 탐지를 넘어 공격의 흐름, 진행 단계, 확산 범위를 파악할 수 있는 시각화 도구를 제공하여 보안 분석 체계를 구축하고자 하였다.

프로젝트의 주요 내용은 다음과 같다. 첫째, Pod 내부에서 발생하는 악성 행위를 시간순으로 추적하는 기능을 구현하였다. 이를 위해 Falco를 사용하여 시스템콜 기반 보안 이벤트를 수집하고, 공격자가 Pod 내부에서 수행할 수 있는 정찰, 권한 확인, 파일 접근, 자격 증명 탐색 등의 행위를 공격 단계로 모델링하였다.

둘째, Pod와 Pod 사이의 네트워크 연결을 분석하여 공격 전파 가능성을 추적하였다. Cilium Hubble을 활용하여 Kubernetes 클러스터 내부의 네트워크 통신 이벤트를 수집하고, 특정 Pod에서 다른 Pod로 이어지는 연결 관계를 분석하였다. 단순히 두 Pod가 통신했다는 사실만으로 감염 여부를 판단하지 않고, 출발지 Pod의 위험 상태, 기존에 없던 새로운 연결 여부, 연결 이후 도착지 Pod의 의심 행위 발생 여부를 종합적으로 고려하였다.

셋째, 서로 다른 형식의 로그를 공통 심볼로 정규화하고, NFA 기반 상태 전이 모델을 적용하였다. Falco와 Hubble에서 수집되는 이벤트는 구조와 의미가 다르기 때문에 원본 로그만으로 직접적인 상관분석을 수행하기 어렵다. 따라서 각 이벤트를 분석 가능한 공통 심볼로 변환하고, Pod별 상태 기계를 통해 공격 진행 상태를 관리하였다. NFA 방식을 적용함으로써 여러 공격 경로를 동시에 유지하고, 중간 이벤트가 존재하더라도 조건에 따라 공격 흐름을 추적할 수 있도록 하였다.

넷째, 분석 결과를 Prometheus와 Grafana를 통해 시각화하였다. 사용자는 대시보드를 통해 각 Pod의 상태, 공격 단계, Pod 간 전파 관계를 확인할 수 있다. 이를 통해 보안 이벤트를 단순 로그 형태로 확인하는 것이 아니라, 공격 체인과 영향 범위를 직관적으로 파악할 수 있도록 하였다.

다섯째, 최종 결과물을 Helm Chart 형태로 패키징하여 배포 가능성을 확보하였다. poding-detector, exporter, Grafana dashboard를 하나의 설치 패키지로 구성하고, GitHub Pages를 통해 설치 방법과 사용 방법을 제공함으로써 다른 Kubernetes 환경에서도 Pod-ing을 설치하고 활용할 수 있도록 하였다.

본 프로젝트를 통해 팀은 Kubernetes 보안 구조, 시스템콜 기반 탐지, 네트워크 플로우 분석, 로그 정규화, 상태 기반 탐지 모델, Prometheus/Grafana 기반 모니터링, Helm Chart 패키징 등 클라우드 네이티브 보안 시스템 개발에 필요한 전반적인 기술을 학습하고 구현 경험을 확보하였다.

4. 시스템 구성 및 내용

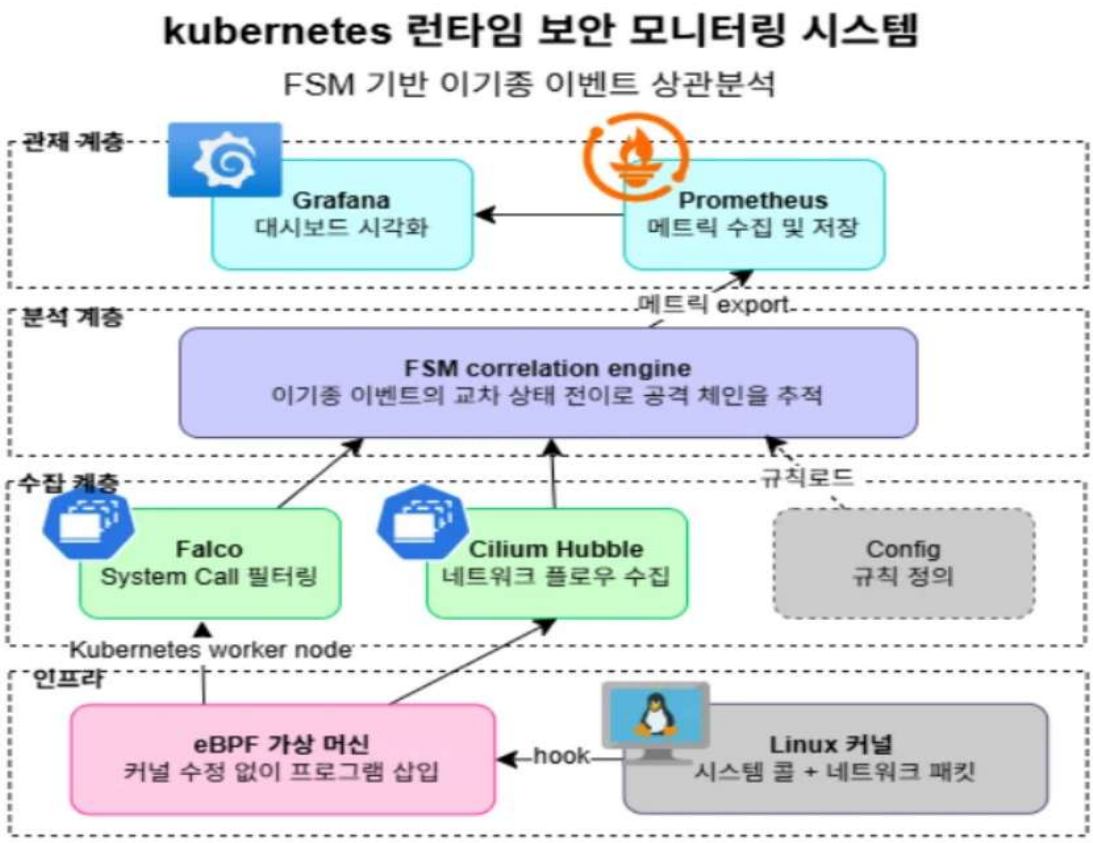


그림 1. Pod-ing 전체 시스템 아키텍처

Pod-ing 시스템은 Kubernetes 클러스터 내부에서 발생하는 시스템콜 이벤트와 네트워크 이벤트를 수집하고, 이를 상관분석하여 Pod 단위 공격 흐름을 추적하는 구조로 설계되었다.

시스템의 동작 과정은 다음과 같다. 먼저 Falco는 각 Pod 내부에서 발생하는 시스템콜 기반 보안 이벤트를 수집한다. 동시에 Cilium Hubble은 Kubernetes 클러스터 내부에서 발생하는 Pod 간 네트워크 통신 이벤트를 수집한다. 수집된 두 종류의 이벤트는 poding-detector로 전달된다.

poding-detector는 서로 다른 로그 형식을 공통 분석 형식으로 변환한다. 이후 상관분석 엔진은 NFA 기반 상태 전이 모델을 이용해 각 Pod의 공격 진행 단계를 추적한다. 공격 단계는 IDLE, RECON, CRED, LATERAL, ALERT로 구성되며, Pod 내부 행위와 Pod 간 연결 이벤트를 함께 고려하여 상태를 전이한다.

특히 Pod 간 통신이 발생했을 때, 시스템은 연결된 Pod를 즉시 감염으로 판단하지 않는다. 출발지 Pod가 이미 위험 상태인지, 기존에 없던 새로운 연결이 발생했는지, 연결 이후 도착지 Pod에서 의심 행위가 발생했는지를 함께 확인한다. 이러한 조건 기반 판단을 통해 정상 통신으로 인한 오탐을 줄이고, 실제 공격 전파 가능성이 높은 흐름을 선별한다.

분석 결과는 exporter를 통해 Prometheus가 수집 가능한 메트릭 형태로 변환되며, Prometheus에 저장된 데이터는 Grafana 대시보드에서 시각화된다. 사용자는 Grafana를 통해 Pod별 위험 상태, 공격 단계, Pod 간 전파 관계를 확인할 수 있다.

5. 프로젝트 결과물에 대한 기술

본 프로젝트의 최종 결과물은 Kubernetes 환경을 위한 이기종 상관분석 보안 솔루션 "Pod-ing"이다. Pod-ing은 Falco와 Cilium Hubble을 활용하여 시스템콜 이벤트와 네트워크 이벤트를 동시에 수집하고, 이를 공통 심볼로 변환한 뒤 NFA 기반 상태 전이 모델을 통해 공격 흐름을 분석한다.
구현 결과, Pod-ing은 Pod 내부 악성 행위 추적과 Pod 간 외부 전파 추적을 모두 수행할 수 있었다. Pod 내부 공격의 경우, 공격자가 침입한 Pod 내부에서 수행하는 정찰 및 의심 행위를 시간순으로 추적하였다. Pod 간 전파의 경우, Pod A에서 Pod B로, 다시 Pod B에서 Pod C로 이어지는 연결 관계를 하나의 공격 체인으로 분석할 수 있도록 하였다.

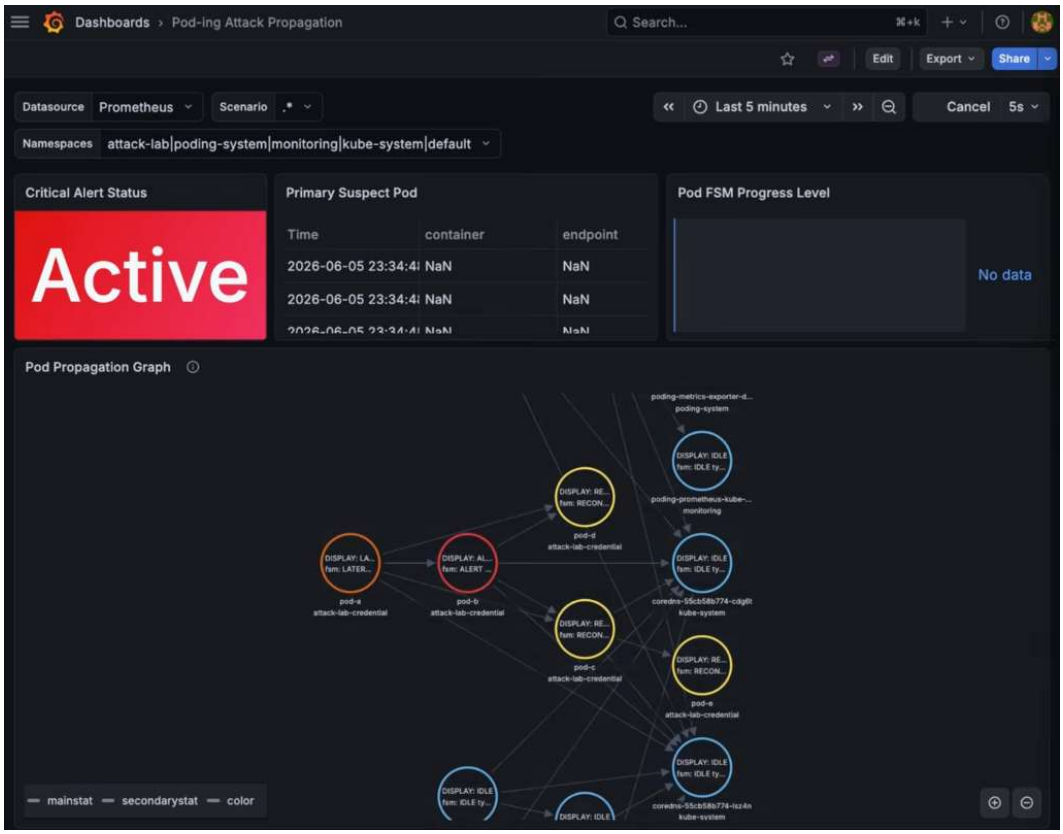


그림 2. Pod-ing의 공격 판정 근거 시각화

그림 2는 Pod-ing이 공격 판정을 내리는 과정을 Grafana 대시보드에서 시각화한 화면이다. 먼저 pod-a 내부에서 shell 실행, token 접근, Kubernetes API 접근과 같은 의심 행위를 시간순으로 추적하고, 이후 pod-a에서 pod-b로 이어지는 네트워크 전파 흐름을 확인한다. Pod-ing은 이 과정에서 단순히 두 Pod가 통신했다는 사실만으로 공격을 판단하지 않고, cross-layer feedback 구조를 통해 Pod 내부 런타임 이벤트와 Pod 간 네트워크 전파 이벤트를 함께 분석한다. 또한 전파를 받은 pod-b에서 후속 의심 행위가 발생하는지까지 확인하여, 내부 행위와 네트워크 전파가 연결된 하나의 공격 체인으로 판단한다.

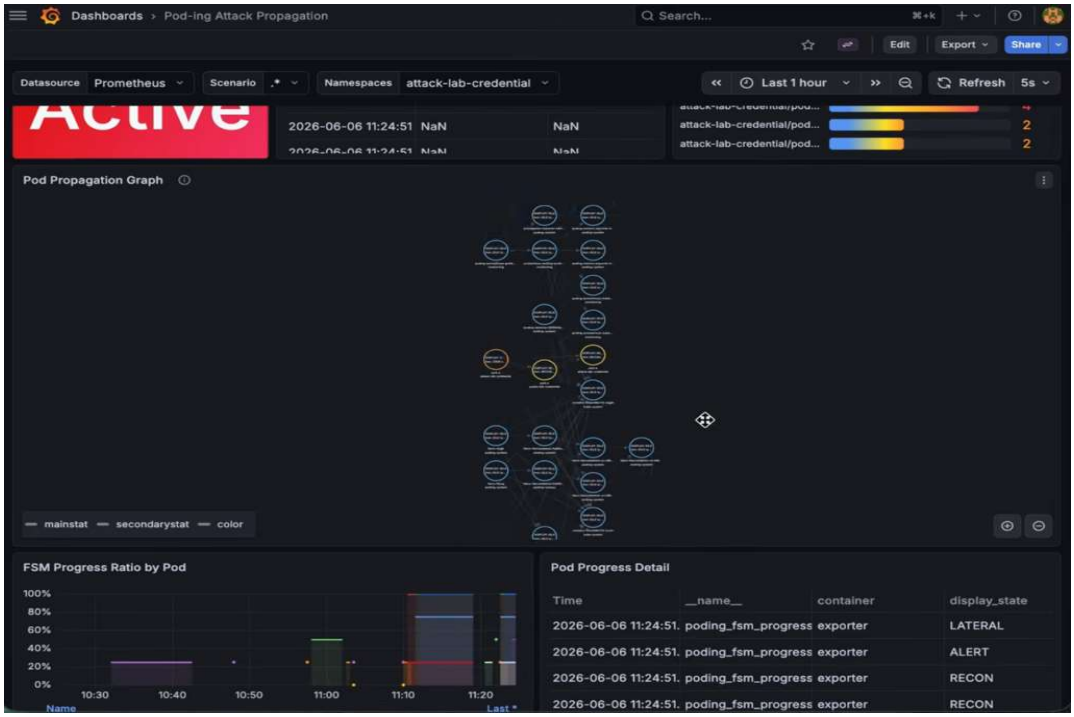


그림 3. 다중 Pod 공격 전파 흐름 동시 추적

그림 3은 하나의 시작점에서 여러 Pod로 공격 흐름이 확산되는 상황을 동시에 추적한 결과이다. Pod-ing은 각 Pod별 FSM 진행 상태를 유지하면서 Pod 간 연결 관계와 각 Pod 내부에서 발생한 후속 행위를 병렬로 분석한다. 이를 통해 여러 공격 경로가 동시에 발생하더라도 각각의 흐름을 유지할 수 있으며, 최종적으로 공격이 어느 Pod까지 확산되었는지를 한 화면에서 확인할 수 있다.

실험은 Falco 단독, Hubble 단독, Pod-ing을 비교하는 방식으로 진행되었다. 총 90개 샘플을 기준으로 Precision, Recall, F1 점수와 오탐률을 측정하여 각 방식의 탐지 성능을 비교하였다.

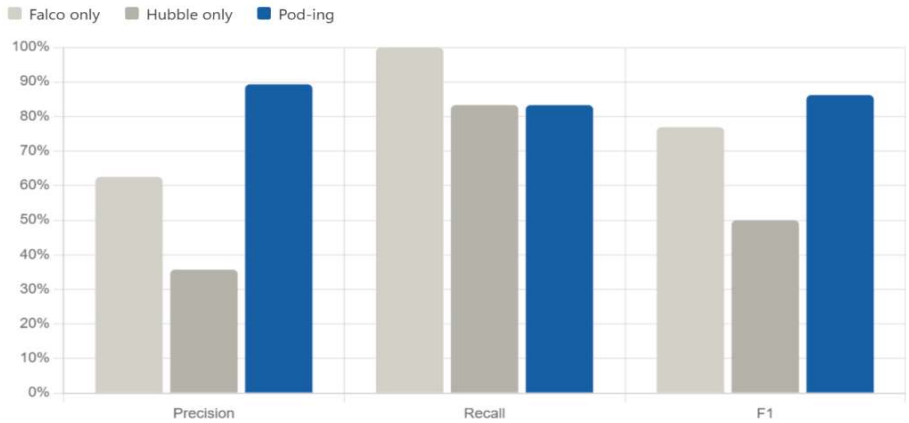


그림 4. Falco, Hubble, Pod-ing의 Precision, Recall, F1 성능 비교

그림 4는 Falco 단독, Hubble 단독, Pod-ing의 Precision, Recall, F1 성능을 비교한 결과이다. Falco는 Pod 내부 시스템콜 이벤트를 세밀하게 관찰할 수 있지만 네트워크 전파 흐름을 함께 해석하는 데 한계가 있었고, Hubble은 Pod 간 통신 관계를 확인할 수 있지만 통신의 원인이 되는 Pod 내부 행위를 파악하기 어려웠다. 반면 Pod-ing은 Falco와 Hubble의 이기종 이벤트를 공통 심볼로 변환하고 NFA 기반으로 상관분석함으로써 단일 도구 기반 탐지보다 높은 Precision과 F1 점수를 보였다. 이를 통해 시스템콜 이벤트와 네트워크 이벤트를 결합한 방식이 Kubernetes 환경의 공격 흐름 탐지에 효과적임을 확인하였다.

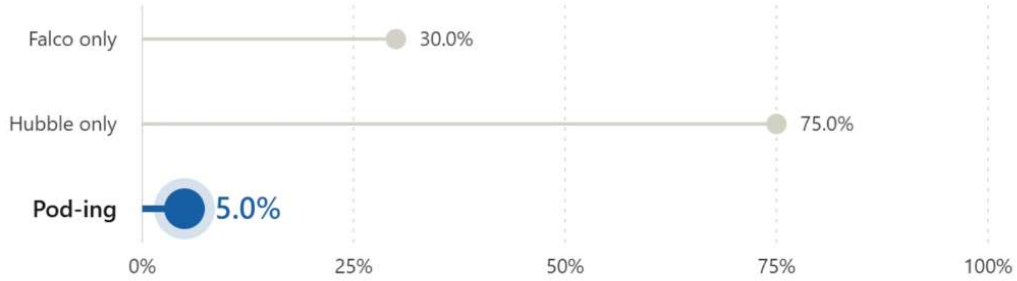


그림 5. Falco, Hubble, Pod-ing의 오탐률 비교

그림 5는 각 방식의 오탐률을 비교한 결과이다. Falco 단독 방식은 일부 정상 행위를 공격으로 판단하여 30%의 오탐률을 보였고, Hubble 단독 방식은 정상 네트워크 통신까지 공격으로 판단하는 경우가 많아 75%의 오탐률을 보였다. 반면 Pod-ing은 Pod 내부 행위와 Pod 간 전파 조건을 함께 고려함으로써 오탐률을 5%까지 낮추었으며, F1 점수는 86.2%로 확인되었다. 이는 단일 로그 기반 탐지보다 시스템콜과 네트워크 이벤트를 함께 분석하는 상관분석 방식이 Kubernetes 환경에서 더 신뢰도 높은 탐지 결과를 제공할 수 있음을 보여준다.

또한 정상 트래픽, 공격과 유사한 정상 행위, Pod 내부 공격, Pod 간 전파 실험을 포함한 자체 벤치마크를 수행하였다. 정상 트래픽 실험에서는 오탐이 발생하지 않았고, Pod 내부 공격 시나리오에서는 기대한 이벤트가 모두 수집되었다. Pod 간 전파 실험에서도 공격 흐름이 성공적으로 탐지되었으며, Grafana UI를 통해 pod-a에서 pod-b로 이어지는 전파 흐름과 여러 Pod로 확산되는 다중 전파 흐름을 확인할 수 있었다.

추가적으로 LID-DS 2021 외부 공개 데이터셋을 활용하여 자체 실험 환경 외의 공격 데이터에서도 분석 방식이 동작하는지 검증하였다. 이를 통해 Pod-ing이 특정 실험 환경에만 국한되지 않고, 다양한 공격 시나리오에 적용될 가능성이 있음을 확인하였다.

최종적으로 Pod-ing은 실험용 코드에 머무르지 않고 Helm Chart 기반 설치 패키지로 구성되었다. poding-detector, exporter, Grafana dashboard를 통합하여 사용자가 자신의 Kubernetes 클러스터에 설치할 수 있도록 하였으며, GitHub Pages를 통해 설치 명령어와 사용 방법을 제공하였다.

구분	기능정의	세부기능 설명
1	이기종 보안 이벤트 수집 기능	Kubernetes 환경에서 발생하는 Pod 내부 시스템콜 이벤트와 Pod 간 네트워크 통신 이벤트를 수집하는 기능이다. Falco를 활용하여 Pod 내부의 시스템콜 기반 보안 이벤트를 수집하고, Cilium Hubble을 활용하여 클러스터 내부의 네트워크 플로우 정보를 수집한다.
2	로그 정규화 및 공통 심볼 변환 기능	Falco와 Hubble에서 수집되는 로그는 형식과 의미가 서로 다르기 때문에 이를 상관분석에 활용할 수 있도록 공통 형식으로 변환하는 기능이다. 각 이벤트를 공격 단계 분석에 필요한 심볼로 변환하여 Pod별 상태 추적이 가능하도록 한다.
3	NFA 기반 공격 흐름 상관분석 기능	정규화된 이벤트를 바탕으로 각 Pod의 공격 진행 상태를 추적하는 기능이다. NFA 기반 상태 전이 모델을 적용하여 IDLE, RECON, CRED, LATERAL, ALERT 단계로 공격 흐름을 분석하며, 중간 이벤트나 여러 공격 경로가 존재하는 상황에서도 유연하게 탐지할 수 있다.
4	Pod 간 공격 전파 추적 기능	Pod 내부 행위와 Pod 간 네트워크 연결을 함께 분석하여 공격이 어느 Pod에서 시작되어 어디까지 확산되었는지 추적하는 기능이다. 출발지 Pod의 위험 상태, 기존에 없던 신규 연결 여부, 도착지 Pod의 의심 행위 발생 여부를 종합적으로 판단하여 공격 전파 가능성을 분석한다.
5	탐지 결과 시각화 및 배포 기능	분석 결과를 Prometheus와 Grafana를 통해 시각화하여 사용자가 Pod의 위험 상태와 공격 체인을 확인할 수 있도록 하는 기능이다. 또한 poding-detector, exporter, Grafana dashboard를 Helm Chart 형태로 패키징하여 다른 Kubernetes 환경에서도 설치 및 활용할 수 있도록 구성하였다.

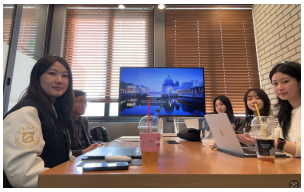
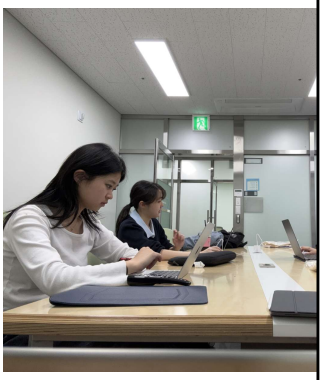
6. 프로젝트 진행내용

1) 참여인원 및 담당 역할

연번	소속학과	성명	수행역할 분담내용
1	인공지능학부	나예원	Kubernetes 실습환경 구축 가이드를 작성하고 팀원들이 동일한 환경에서 클러스터를 구성할 수 있도록 초기 환경 세팅을 주도하였다. 이후 HTTP API 서버 내장, Hubble 연결 폴백 체인 구성, 결과 파일 저장 구조 정비 등 시스템 통합 기능을 담당하였다. 중간 이후에는 Grafana 기반 관제 계층을 재구성하고, Pod 간 공격 전파 관계를 NodeGraph 형태로 확인할 수 있도록 Grafana 연동 API와 dashboard 리소스를 추가하였다. 또한 MITRE 시나리오 검증 및 partial 원인 분석, Pod 내부 benchmark 결과 정리, 최종 발표 슬라이드 구성 등을 수행하였다.

2	인공지능학부	김태림	초기 클러스터 환경 구축에 참여하고, 다단계 풀체인 공격 시나리오를 담당하여 공격 흐름을 설계하였다. 이후 여러 공격 시나리오를 병렬로 추적하는 상태 기계를 구성하고, rule chain 완성 기반 탐지 판정 로직을 설계하였다. Falco와 Hubble 이벤트의 도착 순서가 실제 발생 순서와 달라지는 문제를 해결하기 위해 adaptive delay 로직을 구현·개선하였으며, DelayEstimator와 PodFSM을 연동하여 sensor inversion으로 인해 공격 체인이 끊기지 않도록 보완하였다. 후반에는 MITRE 30개 시나리오·90회 실행 자동화 파이프라인을 구축하고 종합 리포트를 작성하였으며, Helm Chart 기반 배포 구조와 Grafana/Hubble UI 시각화 구성을 정리하였다.
3	인공지능학부	양시은	초기 클러스터 환경 구축에 참여하고, 크립토마이닝 및 권한 상승 공격 시나리오를 담당하여 공격 흐름과 실험 재현 방법을 정리하였다. 프로젝트 중반에는 파이프라인 안내서, 운영 가이드, 심볼 매핑 평가 문서, 시나리오 검증 절차서 등 주요 문서 작성과 검증 절차 정리를 수행하였다. 이후 외부 inbound 트래픽을 침해 시작점 후보로 보고 연결된 Pod의 상태 변화를 함께 추적할 수 있도록 탐지 흐름을 수정하였다. 또한 공개 데이터셋 offline replay와 MITRE ATT&CK 기반 K8s lab 재현 전략을 비교 조사하고, Pod-to-Pod lateral movement 및 cross-layer feedback 구조를 정리하였다.
4	인공지능학부	조예지	초기 Kubernetes 클러스터 환경 구축에 참여하고, 컨테이너 탈출 공격 시나리오를 담당하여 공격 흐름과 실험 재현 방법을 문서화하였다. 이후 Falco와 Hubble의 이기종 이벤트를 하나의 내부 객체로 정규화하고, 심볼 매핑 테이블을 외부 설정 파일로 분리하는 작업을 수행하였다. 또한 공격 시나리오별 Falco 룰과 Hubble flow를 정리하고, 탐지 이벤트가 안정적으로 심볼로 매핑되도록 정규화 로직을 보완하였다. 후반에는 fine symbol 기반 v2 구조를 설계하고, LID-DS 2021 외부 데이터셋 검증 및 전체 실험 결과 정리를 담당하였다.

2) 회의 및 SW멘토링 진행

번호	일시/장소	회의/멘토링 내용(상세히 작성)	관련 사진
1	2026. 4. 1. (16:00~18:00) / 도서관 별관 그룹스터디룸 01호	- Kubernetes 클러스터 실습 환경 구축 방향 논의 - Ubuntu 22.04 기반 VM 2대를 활용하여 컨트롤 플레인 1대, 워커 노드 1대 구조로 클러스터 구성 - kubeadm, containerd, Cilium, Hubble, Falco, Falcosidekick, Kubernetes Dashboard 설치 및 설정 진행 - Hubble UI 및 Falco 이벤트 수집 환경을 구성하고, StorageClass 미설정으로 인한 PVC 바인딩 문제 등 초기 환경 구축 이슈 해결 - 프로젝트에서 다룰 공격 시나리오 6가지 선정: 크립토마이닝, 컨테이너 탈출, Lateral Movement, 권한 상승, 데이터 유출, 다단계 풀체인	
2	2026. 4. 8. (16:00~18:00) / 전남대학교 AI융합대학 3층 스터디라운지 내 스터디룸 1번	- 그동안 진행된 연구 개발 현황과 단계별 구체화 프로세스를 공유하고, 향후 마일스톤 달성을 위한 세부 실행 계획의 보완점 및 리스크 관리 방안에 대해 심층적으로 논의하는 시간을 가짐. - 정기 발표 및 최종 성과 공유회를 대비하여 제한된 시간 내에 프로젝트의 핵심 가치를 명확히 전달할 수 있는 슬라이드 레이아웃 배치, 가시성 높은 데이터 시각화 방안, 그리고 청중을 설득할 수 있는 피치(Pitch) 가이드라인을 전수받음.	
3	2026. 4. 15. (16:00~18:30) / 도서관 정보마루 프레젠테이션룸 03호	- 단일 이벤트 기반의 stateless 탐지 한계를 보완하기 위해 Pod별 이벤트 시퀀스를 축적하는 stateful 탐지 구조 논의 - Falco 이벤트와 Hubble 이벤트를 단일 문자 심볼로 추상화하는 이벤트 심볼 체계 설계 - Cryptomining, Container Escape, Lateral Movement, Privilege Escalation, Data Exfiltration 등 주요 공격 유형에 대한 이벤트 시퀀스 패턴 정의 - 모든 Pod를 하나의 통합 오토마타로 추적하는 방식과 Pod별 개별 오토마타를 구성하는 방식을 비교 검토 - GitHub 리포지토리 기반으로 구현 아이디어 초안 공유 및 개발 방향 정리	
4	2026. 5. 6. (16:30~18:20) / 도서관 별관 그룹스터디룸 02호	- 중간발표 이후 탐지 정확도 개선 및 공격 전파 가시화 기능 구현 회의 진행 - Falco 런타임 이벤트와 Hubble 네트워크 이벤트가 실제 발생 순서와 다르게 detector에 도착하는 문제를 확인하고, 이를 보완하기 위한 adaptive delay 로직 수정 방향 논의 - Pod 내부 FSM 결과와 Pod 간 네트워크 연결 정보를 함께 활용하여 공격 전파 관계를 그래프로 표현하는 구조 설계 - 단순히 Pod A와 Pod B가 통신했다는 이유만으로 공격 전파로 판단하지 않고, Pod 내부 행위와 Pod 간 통신 조건을 함께 만족하는 경우에만 전파 edge로 기록하도록 기준 정리 - Grafana NodeGraph 연동용 API와 dashboard 리소스를 추가하고, 별도 웹 UI에서 Pod 간 연결	

		관계, 공격 전파 edge, Pod별 FSM 상태를 확인할 수 있도록 구성	
5	2026. 5. 13. (16:00~18:30) / 도서관 정보마루 프레젠테이션룸 02호	- 심볼 세분화 구조 도입, Grafana 관제 계층 재구성, adaptive delay 개선, 정확도 측정 방안 설계 회의 진행 - 기존 Grafana NodeGraph가 detector의 correlation edge와 raw Hubble flow를 함께 사용하여 클러스터 전체 네트워크 관계도처럼 보이는 문제를 확인하고, 분석 결과 JSON을 기준으로 동작하도록 관제 계층을 재구성 - raw Hubble graph는 troubleshooting endpoint로 분리하여 실제 탐지 결과와 단순 네트워크 흐름이 혼동되지 않도록 개선 - 기존 coarse symbol만으로 탐지 근거를 세밀하게 구분하기 어려운 한계를 보완하기 위해 fine symbol을 추가하는 v2 Primary·Supporting 구조 설계 - Falco와 Hubble 간 sensor 지연 차이로 인해 공격 체인이 끊기지 않도록 DelayEstimator와 PodFSM을 연동하는 adaptive delay tolerance 동적 적용 방식 논의 - 공개 데이터셋 offline replay 방식과 MITRE ATT&CK 기반 K8s lab 재현 방식을 비교하고, Precision, Recall, F1 등 정량 평가 지표 측정 워크플로우 설계	
6	2026. 5. 20. (16:00~18:00) / 도서관 정보마루 프레젠테이션룸 03호	- 공개 데이터셋 및 MITRE ATT&CK Containers 시나리오 기반 외부 검증 결과 분석 회의 진행 - MITRE ATT&CK Containers Matrix에 대응하는 30개 안전 재현 시나리오를 attack-lab 네임스페이스에서 6개 Phase로 나누어 실행하고 탐지 결과 분석 - 초기 실험 결과에서 발생한 partial 및 fail 원인을 분석하고, 재실행을 통해 fail 항목을 해소하며 탐지 기준 보완 - LID-DS 2021 외부 데이터셋의 syscall trace와 packet을 Pod-ing 내부 포맷으로 변환하여 replay 검증을 수행하고, 외부 데이터셋 기반 탐지 가능성 확인 - Stratus 및 Atomic Red Team의 Kubernetes/Container 공격 기법을 재현하여 Pod-ing 탐지 적합성을 분석하고, 일부 control plane API 중심 공격은 audit source가 추가로 필요함을 확인 - MITRE 30개 시나리오·90회 실행 자동화 파이프라인을 구축하고, 종합 리포트를 작성하여 정량 평가 및 반복 실험 기반을 마련	
7	2026. 5. 27. (16:00~18:00) / 도서관 정보마루 프레젠테이션룸 03호	- Pod-ing 탐지 성능의 평가·검증 체계 구축 회의 진행 - Pod 간 Cross-Layer 전파 탐지 메커니즘을 구현하고, 이미 침해된 파드가 기존 baseline에서 벗어난 신규 peer에 연결할 때에만 전파 신호를 주입하도록 게이트 조건을 설계 - 활동 강도별 동작, 음성 사례, 멀티홉 전이성을 결정론적 replay 방식으로 검증하고, 실제 클러스터의 Falco·Hubble 데이터에서 전파 탐지가 발생하는지 end-to-end로 확인	

		- 정상 트래픽 오탐 평가를 위해 라벨 JSON 스키마와 윈도우 단위 evaluator를 설계하고, 데모 애플리케이션과 Locust 부하생성기를 활용하여 정상 트래픽 및 benign-noise 워크로드를 생성 - 정상 트래픽과 benign-noise 워크로드를 NFA에 투입하여 오탐/정탐을 측정하는 벤치마크 파이프라인을 구축하고, 정상 환경에서의 탐지 거동을 정량적으로 평가할 기반 마련 - Pod 내부 단독 공격 벤치마크 시나리오와 라벨을 추가하고, Falco API endpoint 동적 렌더링, 노이즈 벤치마크, 대시보드 및 main 통합 작업 진행 - 외부 검증 확장을 위해 DARPA TC THEIA CDM 포맷을 Pod-ing 내부 포맷으로 변환하는 어댑터, host-lateral 룰, 진단 스크립트를 추가하고 외부 데이터 기반 검증 가능성 검토	

7. 프로젝트 세부일정 및 내용

No.	작업 내용	4월				5월				6월				담당자	비고
		1	2	3	4	1	2	3	4	1	2	3	4		
1	프로젝트 계획서 작성 및 주제 구체화													전원	주제 선정, 추진 배경 정리, 역할 분담
2	Kubernetes 실습 환경 구축 및 보안 도구 세팅													전원	클러스터 구축, Cilium/Hubble, Falco, Dashboard 구성
3	공격 시나리오 설계 및 실험 재현 방법 문서화													전원	Lateral Movement, 데이터 유출, 컨테이너 탈출, 크립토마이닝, 권한 상승 등
4	이벤트 심볼 체계 및 FSM/NFA 기반 탐지 구조 설계													전원	Falco/Hubble 이벤트 심볼화, 상태 전이 구조 설계
5	이기종 로그 정규화 및 상관분석 엔진 구현													전원	공통 로그 스키마, 심볼 매핑, NFA 기반 탐지 엔진 구현
6	Pod 간 공격 전파 추적 및 Cross-Layer 분석 기능 구현													전원	Pod 내부 행위와 네트워크 전파 관계 결합 분석
7	Grafana 기반 시각화 및 대시보드 구성													전원	NodeGraph, 전파 edge, Pod 상태 시각화
8	탐지 성능 평가 및 외부 데이터셋 검증													전원	MITRE ATT&CK, LID-DS, DARPA TC THEIA, 정상 트래픽 FP 벤치마크
9	Helm Chart 기반 배포 패키지 구성													전원	detector, exporter, Grafana dashboard 패키징
10	최종 발표자료 및 결과보고서 작성·제출													전원	실험 결과 통합, 발표 슬라이드 구성, 결과보고서 작성

8. 결과물에 대한 향후 활용계획

Pod-ing은 향후 Kubernetes 기반 서비스 운영 환경에서 보안 모니터링 및 침해 사고 분석 도구로 활용될 수 있다. 특히 여러 Pod가 동시에 동작하는 클러스터 환경에서 개별 보안 이벤트를 단편적으로 확인하는 것이 아니라, 공격의 진행 과정과 전파 범위를 함께 파악할 수 있기 때문에 보안 담당자의 사고 대응 효율을 높이는 데 기여할 수 있다.

향후에는 실제 운영 환경에서 발생하는 다양한 정상 트래픽과 공격 패턴을 추가로 수집하여 탐지 규칙과 상태 전이 모델을 고도화할 계획이다. 또한 현재의 조건 기반 상관분석 구조를 확장하여 Secret 접근, ServiceAccount 권한 남용, API Server 접근, 클러스터 내부 스캔 행위 등 다양한 Kubernetes 리소스 접근 행위를 추가 분석 대상으로 포함하고자 한다. 이를 통해 Pod 내부 행위와 Pod 간 네트워크 전파를 보다 정밀하게 연결하고, 다양한 공격 시나리오에서도 일반화 가능한 탐지 구조로 발전시킬 수 있다.

시각화 측면에서는 Grafana 대시보드를 개선하여 공격 체인, 영향을 받은 Pod 목록, 시간순 이벤트 흐름, 경보 발생 근거를 더욱 직관적으로 제공할 계획이다. 이를 통해 사용자가 단순히 경보 발생 여부만 확인하는 것이 아니라, 해당 경보가 발생한 이유와 공격이 확산된 범위를 빠르게 이해할 수 있도록 개선할 수 있다. 또한 배포 측면에서는 Helm Chart 패키지를 지속적으로 보완하여 설치 편의성과 다양한 Kubernetes 환경에서의 호환성을 높이고, 장기적으로는 오픈소스 프로젝트 형태로 공개하여 Kubernetes 보안에 관심 있는 사용자와 개발자가 함께 기능을 개선하고 활용할 수 있는 기반을 마련하고자 한다.

더 나아가 본 프로젝트는 1학기 캡스톤 결과물에 그치지 않고, 2학기 후속 연구를 통해 연구 성과로 확장할 계획이다. 추가적인 공격 시나리오와 실제 운영 환경에 가까운 정상 트래픽을 기반으로 실험 검증 범위를 넓히고, 탐지 정확도, 오탐률, F1 점수 등 정량 평가 지표를 보완할 예정이다. 특히 시스템콜 기반 Pod 내부 행위와 네트워크 기반 Pod 간 전파를 함께 고려하는 이기종 상관분석 구조를 심화하여 논문 투고 또는 보안·클라우드 컴퓨팅 분야 학술지 투고를 목표로 한다. 이를 통해 Pod-ing을 단순한 캡스톤 구현 결과물이 아니라 Kubernetes 보안 관측성 및 공격 전파 분석 분야의 연구 기반으로 발전시키고자 한다.

9. 참고자료 및 문헌

1. J. Ryu, R. Kim, S. Lee, S. Kim, H. Choi, and S. Kim, "Hybrid Runtime Detection of Malicious Containers Using eBPF," *Computers, Materials & Continua*, vol. 86, no. 3, p. 13, 2026.
2. F. Wilkens, F. Ortmann, S. Haas, M. Vallentin, and M. Fischer, "Multi-Stage Attack Detection via Kill Chain State Machines," in *Proceedings of the 3rd Workshop on Cyber-Security Arms Race (CYSARM '21)*, 2021.
3. D. Lanoe, M. Hurfin, and E. Totel, "A Scalable and Efficient Correlation Engine to Detect Multi-step Attacks in Distributed Systems," in *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, 2018.
4. H. Kermabon-Bobinnec, Y. Jarraya, L. Wang, S. Majumdar, and M. Pourzandi, "Phoenix: Surviving Unpatched Vulnerabilities via Accurate and Efficient Filtering of Syscall Sequences," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.
5. J. Navarro, A. Deruyver, and P. Parrend, "A Systematic Survey on Multi-step Attack Detection," *Computers & Security*, 2018.
6. Falco 공식 문서, Cloud Native Runtime Security Project.
7. Cilium Hubble 공식 문서, Kubernetes Network, Service & Security Observability.

8. Kubernetes 공식 문서, Pods and Workloads.
9. Prometheus 공식 문서, Monitoring System & Time Series Database.
10. Grafana 공식 문서, Observability and Dashboard Visualization.
11. LID-DS 2021 Dataset, Linux Intrusion Detection Dataset.
12. MITRE ATT&CK, Container and Kubernetes Attack Techniques.
13. CNCF, Cloud Native Security Whitepaper.