KubeSmith: 클라우드 네이티브 환경의 보안 정책을 강화하기 위한 프레임워크

조치현[1], 박현준[2], 이승수[3]

*인천대학교 (대학원생[1], 학부생[2], 교수[3])

KubeSmith: A Framework for Hardening Security Policies in Cloud-Native Environments

CHIHYEON CHO[1], HYEONJUN PARK[2], SEUNGSOO LEE[3]

*Incheon National University

(Graduate student[1], Undergraduate student[2], Professor[3])

요 약

Kubernetes는 클라우드 네이티브 환경에서 컨테이너를 관리하기 위한 핵심 플랫폼으로 널리 활용되고 있다. 이에 따라 KubeArmor, Tetragon과 같은 클라우드 네이티브 런타임 보안 도구들이 사용자 정의 정책을 기반으로 컨테이너 환경을 보호하는 데 사용되고 있다. 그러나 본 연구에서는 이러한 정책들이 공격자에 의해 무력화될 수 있는 두 가지 사례를 식별하였으며, 이를 기반으로 취약한 정책을 자동으로 식별하고 보완할 수 있는 KubeSmith 프레임워크를 제안한다. 또한 KubeSmith의 프로토타입을 구현하고 이를 실험적으로 평가함으로써 그 실현 가능성과 효과성을 입증하였다.

I. Introduction

Kubernetes는 컨테이너화된 워크로드를 관리하기 위한 오픈 소스 컨테이너 오케스트레이션 플랫폼으로, 클라우드 네이티브 환경에서 사실상 표준 인프라로 자리 잡고 있다. 특히 마이크로서비스 아키텍처가 널리 채택됨에 따라, 현대의 IT 기업들은 수많은 컨테이너를 Kubernetes 환경에서 효율적으로 운용하며 서비스의 유연성과 확장성을 확보하고 있다.

그러나 이러한 장점은 보안 측면에서 새로운 공격 지점을 제공하기도 한다. 다양한 오픈 소스 유틸리티와 애플리케이션이 포함된 컨테이너 환경에서는, 공격자가 컨테이너 내부의 파일 시스템에 존재하는 정당한 프로그램을 악용하여 악성 행위를 수행하는 사례가 빈번하다.

이러한 악성 행위를 탐지하고 차단하기 위해, KubeArmor[1]와 Tetragon[2]은 사용자 정의 정책 기반의 런타임 보안 기능을 제공한다.

그러나 본 연구에서는 실험을 통해 이들 보안 정책이 공격자에 의해 무력화될 수 있는 두 가지 사례를 확인하였다. 특히 이러한 사례는 특권이 없는 일반적인 컨테이너 환경에서도 수행할 수 있어, 보안상 심각한 위협이 될 수 있다. 따라서, 본 논문에서는 이러한 무력화 사례를 자동으로 식별하고 완화하는 KubeSmith 프레임워크를 제안한다.

본 논문의 기여 사항은 다음과 같다:

- KubeArmor와 Tetragon에서 제공하는 프로세스 실행 차단 정책이 무력화되는 두 가지 사례를 실험을 통해 식별하고 제시하였다.
- 공격자에 의해 무력화될 수 있는 프로세스 실행 차단 정책을 자동으로 식별하고 강화하는 KubeSmith를 제안하였다.
- 제안한 기법의 실현 가능성을 검증하기 위해 프로토타입을 구현하였으며, 이에 대한 평가를 통해 KubeSmith가 정책 무력화 사례 방지 측면에서 효과적임을 입증하였다.

II. Background

2.1 BPF-LSM

BPF-LSM은 리눅스 커널의 Linux Security Module(LSM) 프레임워크와 eBPF를 결합한 보안 메커니즘이다. LSM은 프로세스 및 파일과 같은 커널 객체에 대한 접근을 제어하기 위해 다양한 hook을 제공하며, 예를 들어 프로세스가 특정 파일에 대해 읽기 작업을 요청할 경우 이들 중 하나인 "file_open" hook이 호출된다.

BPF-LSM은 이러한 LSM Hook에 eBPF 프로그램을 부착함으로써, 접근 요청이 발생할 때마다 해당 요청에 대한 메타데이터를 전달받는다. 이후 eBPF 프로그램은 전달받은 메타데이터를 사용자 정의 정책과 비교하여 해당 요청의 허용 여부를 결정할 수 있다.

2.2 KubeArmor & Tetragon

KubeArmor와 Tetragon은 BPF-LSM을 활용하는 대표적인 클라우드 네이티브 런타임 보안 집행 도구이다. 해당 *시스템 보안 도구들은 관리자 정의 정책을 기반으로, 클라우드에서 발생하는 시스템 호출을 모니터링하거나 제어하는 기능을 제공한다.

Fig 1은 KubeArmor가 제공하는 프로세스 실행 차단 정책을 보여준다. MatchLabels 규칙은 정책이 적용될 대상을 레이블으로 명시하고, Process 규칙은 정책이 프로세스 실행 차단 정책임을 나타낸다. 마지막으로, Path 규칙은 실행 차단 대상 프로그램의 경로를 나타낸다.

```
apiVersion: security.kubearmor.com/v1
kind: KubeArmorPolicy
metadata:
    name: block-package-manager
spec:
  selector:
    matchLabels:
        app: nginx
  process:
    matchPaths:
        - path: /usr/bin/apt-get
    action: Block
```

Fig 1. KubeArmor 정책 예시

* 본 논문에서는 이후의 서술에서 간결함을 위해 KubeArmor와 Tetragon을 통칭하여 '*시스템 보안 도구*'로 지칭한다.

Program Type	Ubuntu	CentOS	Busybox
Package Manager	/usr/bin/apt /usr/bin/apt-get	/usr/bin/yum /usr/bin/rpm	-
File Editor	/usr/bin/vim.basic	/usr/bin/vi	/bin/vi
Add User	/usr/sbin/useradd	/usr/sbin/adduser	/bin/adduser
Shell	/usr/bin/dash	/usr/bin/bash	/bin/sh

Table 1. *시스템 보안 도구*에서 해석하는 컨테이너 별로 상이한 프로그램의 경로

III. Policy Bypass Scenarios

공격자가 일반 권한을 가진 컨테이너를 탈취한 상황을 가정하며, 경로 조작 및 잘못된 경로 구성을 악용하여 *시스템 보안 도구*의 프로세스 실행 차단 정책을 무력화하는 사례를 제시한다.

3.1 경로 조작 기반 보안 정책 우회

*시스템 보안 도구*의 공식 문서 및 예시 정책을 분석한 결과, 이들의 프로세스 실행 차단 정책이 경로 조작 기법에 의해 우회될 수 있는 취약점을 식별하였다. 해당 *시스템 보안 도구*들은 정책에 명시된 경로와 실행 요청된 프로그램의 경로를 비교하여 실행 차단 여부를 결정한다. 그러나 해당 정책은 컨테이너에 존재하는 프로그램의 경로 무결성을 함께 보장하지 않으므로, 공격자는 프로그램 경로를 조작하여 경로 비교 로직을 무력화할 수 있다. 우리는 실험을 통해 경로 변경, 프로그램 복제, 하드링크 생성의 세 가지 방식으로 경로 비교 로직을 무력화할 수 있는 경로 조작 기법을 식별하였다.

3.2 잘못된 경로 구성(Misconfiguration) 악용

*시스템 보안 도구*는 경로 기반으로 프로세스 실행을 차단한다. 하지만, 컨테이너는 독립적인 파일시스템을 가지므로 같은 유형의 프로그램이라도 그 경로가 컨테이너마다 상이하다. 또한 *시스템 보안 도구*들은 프로그램의 경로를 심볼릭 링크의 최종 경로로 해석하며, Table 1은 *시스템 보안 도구*에서 해석하는 컨테이너 이미지별로 상이한 프로그램의 경로를 나타낸다. 이러한 복잡한 특성으로 인해, 해당 *시스템 보안 도구*에서 공식적으로 제공하는 보안 정책을 다양한 컨테이너가 실행되는 클라우드 환경에 적용한 결과, 일부 컨테이너에서는 보안 정책이 올바르게 적용되지 않는 것을 식별하였다.

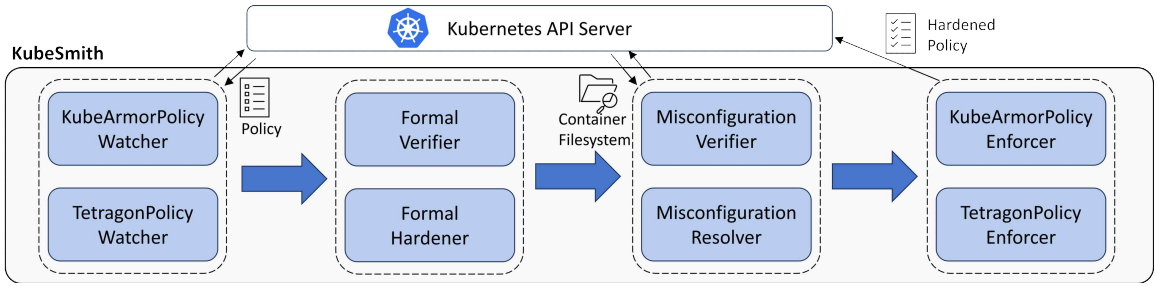

Fig 2. KubeSmith의 아키텍처

Bypass Type	LSM Hook	KubeArmor rule	Tetragon rule
Path Modification	path_rename	File	-
File Duplication	file_open		hook: "file_open"
Hardlink	path_link		-

Table 2. 경로 조작 패턴-대응 규칙 매핑

IV.System Design

4.1 KubeSmith Overview

본 논문은 보안 정책 무력화 공격을 효과적으로 방지하기 위한 KubeSmith 프레임워크를 제안하며, 이는 형식 검증 및 강화 기술과 잘못된 경로 구성 식별 및 수정 기술로 구성된다.

4.2 Formal Verifier and Hardener

본 논문에서는 §3.1에서 제시한 세 가지 경로 조작 기법을 효과적으로 차단하기 위해, *시스템 보안 도구*의 취약한 정책을 식별하고 이를 강화하는 기법을 제안한다. 이는 경로 조작 패턴-대응 규칙 매핑, 취약한 정책 식별, 정책 강화의 세 단계로 구성된다.

경로 조작 패턴-대응 규칙 매핑. Table 2는 공격자가 각 경로 조작 기법을 수행할 때 호출되는 LSM Hook과 이에 대응하는 보안 정책의 규칙을 나타낸다. KubeArmor의 경우 File 규칙을 사용하여 모든 경로 조작 기법을 방지할 수 있는 반면, Tetragon은 각 경로 조작 기법에 따라 상이한 규칙 구성이 필요하다.

형식 검증 기반 우회 취약 정책 식별. Formal Verifier는 Watcher로부터 전달받은 프로세스 실행 차단 정책의 규칙을 분석하여, 차단 대상 프로그램의 경로를 추출한다. 이후 경로 조작 패턴-해당 경로에 대한 경로 조작 방지 규칙이 누락된 취약한 정책을 식별한다.

규칙 보강 기반 정책 강화. Formal Hardener는 Foraml Verifier로부터 전달받은 취약 정책에 대해, 경로 조작 패턴-대응 규칙 매핑을 기반으로 규칙을 보완한다. 이를 통해 해당 정책이 차단하는 프로그램에 대해 경로 조작을 원천적으로 차단할 수 있도록 강화한다.

4.3 Misconfiguration Verifier and Resolver

본 논문에서는 §3.2에서 제시한 보안 정책의 잘못된 경로 구성을 효과적으로 완화하기 위해, 대형 언어 모델(LLM)을 활용해 정책 적용 대상 경로를 검증하고 보정하는 기법을 제안한다. KubeSmith는 컨테이너 파일시스템 정보 추출, 컨테이너-프로그램명 매칭, 파일시스템 조회 기반 경로 보정의 세 단계로 구성된다.

컨테이너 파일시스템 정보 추출. 보안 정책의 MatchLabels 규칙을 파싱하여 클러스터 내에 해당 Label을 가진 Pod 목록을 조회한다. 이후, Pod를 구성하는 컨테이너의 파일시스템에서 리눅스 배포판 정보를 추출한다.

컨테이너-프로그램명 매칭. 제안하는 기술은 LLM이 다양한 리눅스 배포판에서 사용되는 프로그램명 간의 차이를 학습하고 있는 점을 활용한다. 정책 적용 대상 프로그램과 컨테이너에서 추출한 리눅스 배포판 정보를 기반으로 LLM에 입력할 프롬프트를 생성한다. 생성된 프롬프트를 통해 LLM에 질의함으로써, 배포판에 따라 상이할 수 있는 프로그램명을 적절히 보정한 결과를 획득한다.

파일시스템 조회 기반 경로 보정. LLM이 출력한 프로그램명을 기반으로, 대상 컨테이너 내부에서 심볼릭 링크의 최종 경로를 탐색한다. 이를 통해 정책의 잘못 지정된 경로를 올바른 경로로 보정함으로써 정책을 강화한다.

V. Implementation & Evaluation

5.1 Prototype Implementation

본 논문에서는 제안한 기술의 실현 가능성을 검증하기 위한 프로토타입을 구현하였다. 보안 정책의 감지, 수정 및 집행을 위해 Go언어 기반의 client-go 패키지를 활용하였으며, 컨테이너-프로그램 매칭 기능은 ChatGPT를 활용하여 실현 가능성을 평가하였다.

5.2 Evaluation

본 논문은 정책 무력화 방지 측면에서 KubeSmith의 효과성을 평가하였다. 이를 위해, *시스템 보안 도구*의 GitHub 저장소에서 시스템 보안 정책을 각 5개씩 선별하여 평가용 데이터셋을 구축하였다. 또한, *시스템 보안 도구*만 작동하는 Kubernetes 클러스터 환경과 KubeSmith가 함께 작동하는 환경을 구축하고, 각 환경에 해당 데이터셋을 동일하게 적용하였다.

경로 조작 기반 보안 정책 우회 방지. 각 환경에 배포된 임의의 파드에서 정책 적용 대상 프로그램에 대한 경로 조작을 시도하였다. Fig 3은 '/usr/bin/apt' 실행 차단 정책이 적용된 컨테이너에서, 공격자가 경로를 조작해 정책을 우회하는 사례를 나타낸다. Fig 3(A)는 KubeSmith가 적용되지 않은 환경으로, 공격자가 경로 조작을 통해 정책을 우회한 모습을 나타낸다. 반면, Fig 3(B)에서는 KubeSmith가 적용된 환경에서 경로 조작이 성공적으로 차단되어 우회 시도가 실패하는 것을 확인할 수 있다.

하지만, KubeArmor는 모든 경로 조작 시도를 성공적으로 차단한 반면, Tetragon은 파일 복제를 제외한 나머지 경로 조작 시도를 차단하지 못하였다. 이는 Table 2에서 확인할 수 있듯이, Tetragon이 해당 경로 조작 유형을 차단할 수 있는 규칙을 현재 지원하지 않기 때문이다. 향후 Tetragon이 관련 규칙을 지원할 경우, 이는 자연스럽게 해결될 것으로 기대한다.

잘못된 경로 구성 악용 방지. 각 환경에 배포된 정책 데이터셋이 다양한 컨테이너에 정확히 적용되는지 비교하여, 잘못된 경로 구성 식별 및 수정 기능의 효과성을 측정하였다. Table 3은 KubeSmith의 실행 여부에 따른 컨테이너 이미지별 정책 적용 결과를 프로그램 유형별로 나타낸 것이다. 예를 들어, Ubuntu 컨테이너에 File Editor를 차단하는 정책을 적용한 결과, 기본 환경에서는 잘못된 경로 구성으로 인해 1개의 프로그램 중 0개를 차단했지만 KubeSmith를 활용한 경우 컨테이너에 존재하는 프로그램을 정확하게 차단한 것을 확인할 수 있다.

Program Type	Ubuntu		CentOS		Busybox	
	w/o	w/	w/o	w/	w/o	w/
Package Manager (apt)	2/2	2/2	**0/2**	**2/2**	-	-
File Editor (vim)	**0/1**	**1/1**	**0/1**	**1/1**	1/1	1/1
Add User (adduser)	1/1	1/1	1/1	1/1	**0/1**	**1/1**
Shell (sh)	**0/1**	**1/1**	1/1	1/1	1/1	1/1

Table 3. KubeSmith의 잘못된 경로 구성 식별 및 보정 기능의 효과성

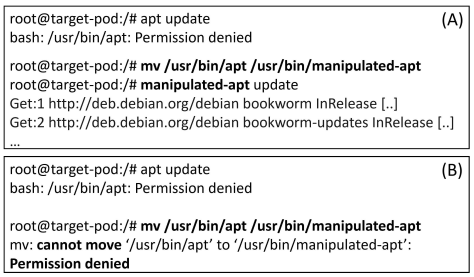

Fig 3. KubeSmith의 경로 조작 기반 보안 정책 우회 방지 Use case

VI. Conclusion

본 논문에서는 실험을 통해 *시스템 보안 도구*의 프로세스 실행 차단 정책을 무력화할 수 있는 사례를 제시하고, 이를 완화할 수 있는 기술을 제안하였다. 또한, 실험을 통해 이들의 실현 가능성과 효과성을 입증하였다.

Acknowledgements

본 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원 - 학·석사연계ICT핵심인재양성 지원을 받아 수행된 연구임(IITP-2025-RS-2024-00437024)

[참고문헌]

[1] KubeArmor, May 2025, [online] Available: https://github.com/kubearmor/KubeArmor

[2] Tetragon, May 2025, [online] Available: https://github.com/cilium/tetragon