Kunerva+: 컨테이너를 위한 지능형 보안 정책 생성 프레임워크

김봄[1], 최유정[2], 이승수[3]

인천대학교 (대학원생[1], 학부생[2], 교수[3])

Kunerva+: An Intelligent Security Policy Geneartion Framework for Containers

Bom Kim[1], YuJung Choi[2], Seungsoo Lee[3]

*Incheon National University

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

요 약

현대 클라우드 네이티브 환경에서 컨테이너 기술은 필수적이지만, 그 복잡성으로 인해 수동 정책 관리 방식으로는 효율성과 일관성을 보장하기 어렵다. 본 논문에서는 사용자의 자연어 입력을 분석하여 네트워크 및 시스템 보안 정책을 자동으로 생성하고 시행하는 지능형 프레임워크 Kunerva+을 제안한다. 이를 통해 보안 정책 설정의 복잡성을 줄이고 인적 오류를 최소화하며, 실시간으로 변화하는 컨테이너 환경에 신속하게 대응할 수 있다. 성능 평가 결과, 미세 조정된 모델이 기본 모델보다 높은 BLEU 점수를 획득했으며, 실제 환경에서의 실험을 통해 정책 생성의 정확성과 효율성을 입증하였으며, 보안 관리에서 LLM의 활용 가능성을 보여준다.

I. Introduction

컨테이너 기술은 애플리케이션 배포와 관리의 효율을 개선하였지만, 동시에 새로운 보안 문제를 야기한다. 컨테이너화된 애플리케이션은 마이크로서비스 아키텍처를 기반으로 운영되며, 개별 컨테이너의 생성과 소멸이 빈번하게 발생한다. 이러한 동적 특성으로 인해 전통적인 방화벽 설정이나 접근 제어 리스트(ACL)로는 대응하기 어렵다. 또한 네트워크 구성이 자주 변경되고 서비스 간의 상호작용이 많아 보안 정책 관리가 복잡해지며, 데이터 유출이나 미승인 접근과 같은 보안 취약점을 초래한다.

컨테이너 환경에서의 보안 정책은 애플리케이션과 데이터를 보호하는 데 중요한 역할을 한다. 그러나 기존의 수동적 보안 정책 관리 방식은 시간 소모가 많고 인적 오류가 발생하기 쉽다. 또한 오류에 동적으로 대응하기 힘들고, 정책의 일관성과 정확성을 보장하기 어렵다.

본 논문에서는 컨테이너화된 환경의 복잡성과 컨텍스트 정보 부족으로 인한 보안 문제를 효과적으로 해결하고자 Kunerva+을 제안한다. Kunerva+는 사용자가 자연어로 보안 의도를 입력할 수 있게 하고, 이를 분석하여 자동으로 네트워크 및 시스템 보안 정책을 시행한다.

본 논문의 주요 기여는 다음과 같다:

1. 자연어 기반 보안 정책 생성: 자연어로 보안 요구를 입력함으로써 사용자 중심의 접근 방식을 통해 보안 요구사항과 실제 보안 정책 사이의 정책 생성 간극을 줄이고 정책 설정의 복잡성을 감소시킨다.

2. 정책 사전 검증 및 자동화: 생성된 보안 정책에 대한 CRD 구문, 리소스 및 속성 검증을 제공하여 잘못 구성된 정책이 시스템에 적용되는 것을 예방하며 신뢰성과 안정성을 보장한다.

II. Background and Related Work

2.1 Security Policy in Cloud-native Environments

Cloud-native 환경에서 보안 정책은 컨테이너의 동작을 제어하고 이상 행위를 차단한다.

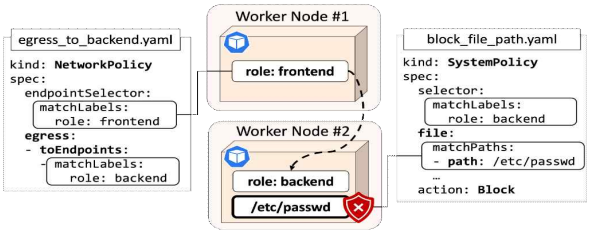

Fig 1. The Example of Security Policy Enforcement

네트워크 정책은 파드 간 또는 네트워크 엔드포인트와의 통신 규칙을 정의하여 트래픽을 관리하고 서비스 간 격리를 강화한다. 시스템 정책은 컨테이너가 호스트 시스템 자원을 어떻게 제어할지를 규정하며, 파일 및 프로세스에 대한 보안 규칙을 설정하여 무단 접근을 차단한다.

Fig 1은 두 개의 Worker Node에 'role: frontend'와 'role: backend' 레이블을 가진 파드가 배포된 상태에서 네트워크 정책과 시스템 정책을 적용한 예시를 보여준다. 첫 번째 시나리오는 frontend 파드에 외부 통신을 허용하는 네트워크 정책을 적용한 경우이다. 이 정책은 backend 파드로의 트래픽을 허용하고 다른 모든 연결을 제한하여 파드 간 통신을 제어하고 불필요한 접근을 차단한다. 두 번째 시나리오는 backend 파드 내 '/etc/passwd' 파일에 대한 접근을 제한하는 시스템 정책을 보여준다. 이 정책을 통해 클러스터는 신뢰할 수 있는 접근만이 파일을 조회할 수 있도록 한다.

2.2 Large Language Model (LLM)

자연어 처리는 컴퓨터가 텍스트와 음성 데이터를 분석, 이해, 생성, 조작할 수 있게 하는 인공지능 분야이다. LLM은 자연어 처리의 핵심 기술로, 대량의 텍스트 데이터로 학습한 수십억 개의 파라미터를 가진 신경망 모델이다. LLM은 텍스트 생성 및 분류와 같은 복잡한 언어 작업에서 높은 성능을 보이며, 사전 훈련된 LLM은 특정 작업을 위해 파인튜닝할 수 있다. 파인튜닝(Fine-tuning)은 작은 데이터 세트로 기존 LLM을 재훈련하여 특정 작업에 맞춰 파라미터를 조정하는 것을 의미하며, 이를 통해 특정 작업의 정확도를 향상할 수 있다.

2.3 Related Work

Jacobs et al. [1]은 자연어로 네트워크 관리 의도를 표현하고, 이를 머신러닝과 오퍼레이터 피드백을 통해 정확하게 해석하여 네트워크 구성 명령으로 변환하는 방법을 제안했다. 또 다른 연구인 Li et al. [2]은 마이크로서비스 간의 접근 제어 정책을 자동 생성하고, 정적 분석 및 그래프 기반의 정책 관리 메커니즘을 사용하여 동적 환경에서도 적시에 정책을 업데이트하는 방법을 제안했다.

기존 연구들이 주로 네트워크 정책에 집중한 반면, 본 연구에서는 LLM을 활용한 보안 관리 프레임워크를 제안하며, 네트워크와 시스템 보안 정책의 통합 관리 및 자동화된 검증 및 시행 메커니즘에 초점을 맞춘다.

III. System Design

3.1 Kunerva+ Design

Kunerva+는 두 가지 요구사항을 충족하도록 설계되었다. 첫째, 사용자의 자연어 입력을 정확하게 분석하여 정책으로 변환해야 한다. 다변적인 컨테이너 환경의 복잡한 컨텍스트를 이해하고 모든 메타데이터와 정보를 분석하여 적절한 정책을 생성하는 것이 중요하다. 둘째, 생성된 보안 정책을 검증하고 적용하는 과정을 자동화해야 한다. 정책의 정확성과 일관성을 보장하고 잘못된 정책 적용을 방지하며 변화하는 보안 요구사항에 대응하기 위해 필수적이다.

Kunerva+의 동작 흐름은 Fig 2와 같다. 사용자가 UI를 통해 자연어로 요구사항을 입력하면, Intent 및 Entity Classifier가 의도를 분석한다. 분석된 정보는 Prompt Processor에서 구체적인 보안 정책 요구사항으로 변환된다. 이 프롬프트는 Policy Generator로 전달되어, 미세 조정된 LLM을 통해 보안 정책으로 변환되고, Policy Processor에 의해 적용할 수 있는 형태로 처리된다. 생성된 정책은 Validator에서 검증을 거쳐 Enforcer를 통해 정책을 적용한다. Kunerva+는 자연어로 의도를 표현하는 것만으로도 복잡한 보안 정책을 관리할 수 있게 한다.

3.2 Policy Creation

Kunerva+는 사용자로부터 자연어 입력을 바탕으로 보안 정책을 생성하기 위해 BERT 기반

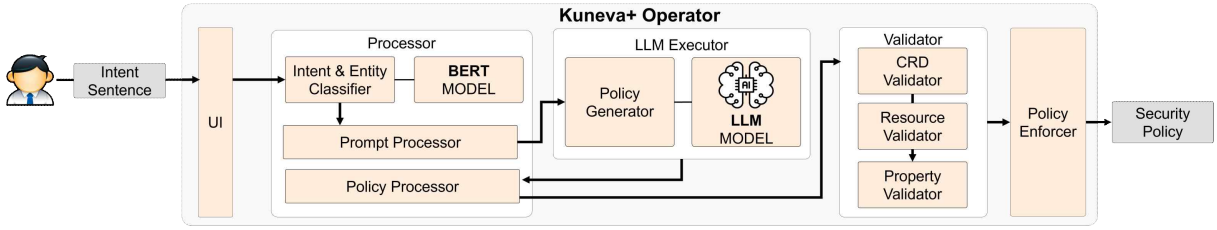

Fig 2. The Architecture of Kunerva+

Table 1. Summary of Datasets

Name	Type	Size	
Network Policy	JSON Lines	166,064개	187.3 MB
System Policy		2,914개	3.27 MB
Network Intent	CSV	1,500개	4.10 MB
System Intent		1,500개	(3000개)

의 Intent 및 Entity Classifier를 활용하여 입력된 문장의 주요 의도가 네트워크 또는 시스템 정책인지를 파악하고, 파드 이름, 파일 경로, 액션와 같은 필요한 엔티티를 식별한다. 추출된 의도와 엔티티 정보를 바탕으로 Prompt Processor는 세부 정보를 포함한 프롬프트로 재구성한다. 구성된 프롬프트를 통해 단순한 정보 이상의 컨텍스트를 반영할 수 있으며 Policy Generator에 전달되어, 미세 조정된 LLM를 통해 실제 보안 정책을 생성한다. 사용자는 Table 2에 표기된 여러 LLM 중 필요한 모델을 선택할 수 있다. 생성된 정책은 Policy Processor를 통해 정책 이름과 같은 메타데이터를 추가한다.

3.3 Policy Validation and Enforcement

Validator의 첫 번째 단계는 Custom Resource Definition(CRD) 검증으로, 정책의 구문을 확인하여 형식적인 오류를 방지한다. 두 번째 단계는 리소스 검증으로, 정책이 참조하는 리소스의 존재를 확인한다. 예를 들어 파드의 존재와 네임스페이스의 활성화 여부를 확인한다. 세 번째 단계는 속성 검증으로, 정책 조건이 실제 클러스터 환경에서 충족되는지 확인한다. 예를 들어 정책에 명시된 파일 경로가 존재하는지, 지정된 포트가 리스닝 되고 있는지 등을 확인한다. 문제가 발생하면 사용자에게 보고하고 완료되면 Enforcer를 통해 정책을 시행한다.

IV. Evaluation

4.1 Test Enviroments

본 실험 환경은 Ubuntu 22.04 기반 가상머신 3대로 쿠버네티스 클러스터를 구축하고 마스터 노드에 Kunerva+ 프레임워크를 배포하였다. LLM 파인튜닝은 GPU NVIDIA A6000 2개가 장착된 서버에서 수행되었으며 실험은 실제 환경에서의 적용 가능성과 파인튜닝 모델의 효용성을 평가하기 위해 진행되었다. 프로토타입은 Go와 Python으로 약 3.3K 줄의 코드로 구현되었으며 Cilium Network Policy와 KubeArmor Policy를 지원한다.

Table 2. Summary of Model Features

Model Name	Size (params)	Purpose
bert-intent-classification	425 MB	Intent Classification
bert-ner-classification	425 MB	NER Classification
Meta/Meta-Llama-3-8B-Instruct	16GB (8.03B)	Policy Generation
DeepSeek/deepseek-coder-7b-instruct-v1.5	14GB (6.91B)	Policy Generation
MistralAI/Mistral-7B-Instruct-v0.2	15GB (7.24B)	Policy Generation
Google/codegemma-7b-it	17GB (8.54B)	Policy Generation
Meta/codeLlama-7b-Instruct-hf	14GB (6.74B)	Policy Generation

4.2 Details for Dataset and Model

Table 1과 같이, 데이터셋은 Github에서 오픈된 정책 파일을 스크랩한 후, 필드 값을 수정하고 Selector와 Rule을 분할하여 Instruction을 생성하는 방식으로 증식하였다. 추가로 Instruction을 이용하여 네트워크와 시스템의 Intent 데이터셋도 함께 생성하였다.

Table 2처럼, 분류 모델은 BERT 기반으로 임베딩 모델을 고정하고 커스텀한 classifier로 설계하였다. Intent 모델은 네트워크와 시스템 두 가지 타입을, Entity 모델은 13가지 타입을 분류한다. 정책 생성에는 text-to-text 디코더 전용 오픈 모델을 기준으로 총 5개의 모델을

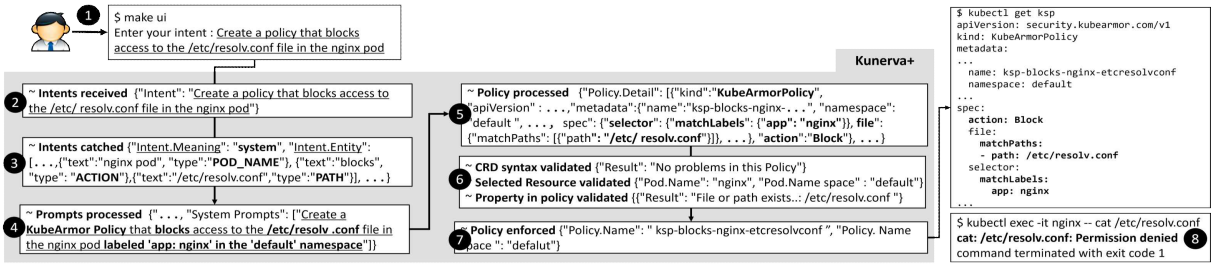

Fig 3. Workflows of Kunerva+

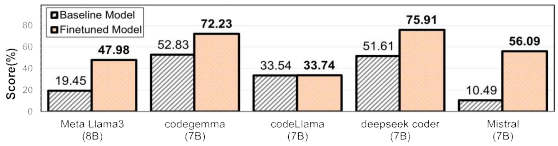

Fig 4. Comparison of BLEU Score for baseline and fine-tuned models

선정하였다. 선정 기준은 다음과 같다. :

1) 적은 데이터셋으로도 높은 성능을 발휘하기 위해 작은 크기와 적은 파라미터 수로도 뛰어난 정확도를 제공하는 모델

2) 입력된 생성 요청 Instruction에 대해 평균 이상의 Policy 출력을 가지는 모델

4.3 Use Case

Fig 3과 같이, 파일 액세스를 제한하는 시나리오로 Kunerva+의 효율성을 평가하였다. 사용자가 'Create a policy that blocks access to the /etc/resolv.conf file in the nginx pod'와 같이 정책을 요청한다(①). 시스템은 이 입력을 받아 문장을 분석하고(②) 의도와 엔티티를 추출한다(③). 요청은 시스템 정책으로 분류되며, 파드 이름, 액션, 파일 경로가 추출된다. 이를 바탕으로 프롬프트가 재구성되고(④), 보안 정책이 생성된다(⑤). 변환된 정책에 대해 CRD 검증에서는 문법적 정확성을, 리소스 검증에서 파드와 네임스페이스의 존재를, 속성 검증에서 파일 경로의 존재를 확인한다(⑥). 모든 검증이 완료되면 정책은 실제 컨테이너 환경에 적용된다(⑦). 이후 파일에 대한 접근을 시도하면 'Permission denied'가 반환된다(⑧). 이는 Kunerva+을 통해 생성된 정책이 성공적으로 적용되었음을 증명하며 실제 환경에서 효과적으로 보안 정책을 시행할 수 있음을 입증한다.

4.4 Performance

Fig 4와 같이, Kunerva+의 정책 생성 성능을 평가하기 위해 기본 모델과 미세 조정된 모델의 BLEU 점수를 비교하였다. Meta Llama3 모델의 경우 기본 모델의 정확도는 19%였으나 미세 조정된 모델은 47%로 크게 향상되었고, 다른 모델들 또한 유사한 경향을 보였다. 이는 미세 조정된 모델이 기본 모델보다 효과적으로 보안 정책을 생성할 수 있음을 시사한다. Kunerva+의 모델 최적화를 통해 보안 정책 생성의 정확성을 향상했으며 보안 관리에서의 활용 가능성을 입증한다.

V. Conclusion and Future Work

Kunerva+는 보안 정책 생성 및 검증을 통합하여 네트워크 및 시스템 정책 집행을 단순화한다. 실험을 통해 Kunerva+가 실제 환경에서 실현할 수 있으며, LLM을 보안 관리에서 활용할 수 있음을 입증하였다.

향후 연구에서는 오로지 리소스 구성 파일을 바탕으로 사용자의 의도를 유추하여 최적의 보안 정책을 생성하는 연구를 진행할 예정이다.

[참고문헌]

[1] Jacobs, Arthur S., et al. "Hey, lumi! using natural language for {intent-based} network management." 2021 USENIX Annual Technical Conference (USENIX ATC 21). 2021.

[2] Li, Xing, et al. "Automatic policy generation for {Inter-Service} access control of microservices."30th USENIX Security Symposium (USENIX Security 21). 2021.