Confiler: 컨테이너 환경을 위한 지능형 파일 접근 제어 프레임워크

박현준[1], 이승수[2]

*인천대학교 (학부생[1], 교수[2])

Confiler: An Intelligent File Access Control Framework for Containerized Environments

Hyeonjun Park[1], Seungsoo Lee[2],

*Incheon National University

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

요 약

컨테이너는 경량화된 배포와 빠른 실행이 가능해 다양한 서비스 환경에서 널리 활용되고 있다. 그러나 서비스 확산과 함께 런타임 보안의 중요성도 커지고 있으며, 특히 파일 시스템 접근 제어는 여전히 주요 취약 지점으로 남아 있다. 기존 블랙리스트 기반 정책은 새로운 공격에 대응하기 어렵고, 화이트리스트 정책 생성을 위한 테스트 케이스 생성 자동화 기법은 낮은 커버리지와 높은 수동 개입으로 효율성이 떨어진다. 또한 동적으로 생성되는 파일 경로를 적절히 반영하지 못하는 정책은 정상 접근을 차단하거나 공격 표면을 확대할 수 있다. 이에 본 논문에서는 large language model(LLM)을 활용해 테스트 케이스를 자동 생성하고, 실행 과정에서 관찰된 파일 접근 경로를 분석하여 동적 경로를 식별함으로써 화이트리스트 정책을 자동 생성하는 Confiler 프레임워크를 제안한다. 또한 Confiler의 프로토타입을 구현하고 이를 실험적으로 평가함으로써 그 효과성도 입증하였다.

I. Introduction

컨테이너 기술은 애플리케이션을 독립된 환경에서 실행할 수 있도록 하여, 경량화된 배포와 빠른 실행을 가능하게 한다. 이러한 특성으로 인해 컨테이너는 마이크로서비스 환경에서 널리 활용되고 있다. 그러나 보안 측면에서는 여전히 취약한 부분이 존재하며, 특히 파일 시스템 접근 제어는 컨테이너 보안의 핵심 요소임에도 불구하고 근본적인 한계를 지닌다.

기존 블랙리스트 기반 정책은 알려진 위협만 차단하므로 새로운 공격이나 Zero-day 취약점에는 대응이 어렵다. 또한 화이트리스트 정책 생성을 위한 테스트 케이스 자동화 기법은 커버리지가 낮고 수동 개입이 많아 마이크로서비스 환경에 적합하지 않다. 더불어 동적 경로를 반영하지 못하는 정책은 정상 접근을 차단하거나 과도한 일반화로 보안성을 저하시킬 수 있다.

본 논문에서는 이러한 한계를 해결하기 위해 Confiler라는 화이트리스트 정책 자동 생성 프레임워크를 제안한다. Confiler는 LLM을 활용하여 프로그램의 분기 경로를 포괄하는 테스트 케이스를 자동 생성하고, 실행 중 관찰된 파일 접근 경로를 분석하여 동적 경로를 식별하고 이를 반영한 정책을 생성한다.

본 논문의 기여 사항은 다음과 같다:

- Confiler는 실제 소스코드를 기반으로 LLM을 활용하여 테스트 케이스를 자동 생성함으로써 파라미터 입력 정확도를 향상시켰다.
- 제안한 기법의 실현 가능성을 검증하기 위해 프로토타입을 구현하였으며, 평가를 통해 Confiler가 테스트 커버리지 측면에서 기존 자동화 도구보다 효과적임을 입증하였다.

II. Background & Motivation

2.1. Background

컨테이너는 애플리케이션과 실행 환경을 함께 패키징하여, 경량화된 배포와 격리된 실행 환경을 제공한다. 이러한 특성은 마이크로서비스 아키텍처(MSA)의 확산을 이끌었다. MSA는 애플리케이션을 독립적인 서비스 단위로 구성함으로써 유연성과 확장성을 높인다. 한편, 컨

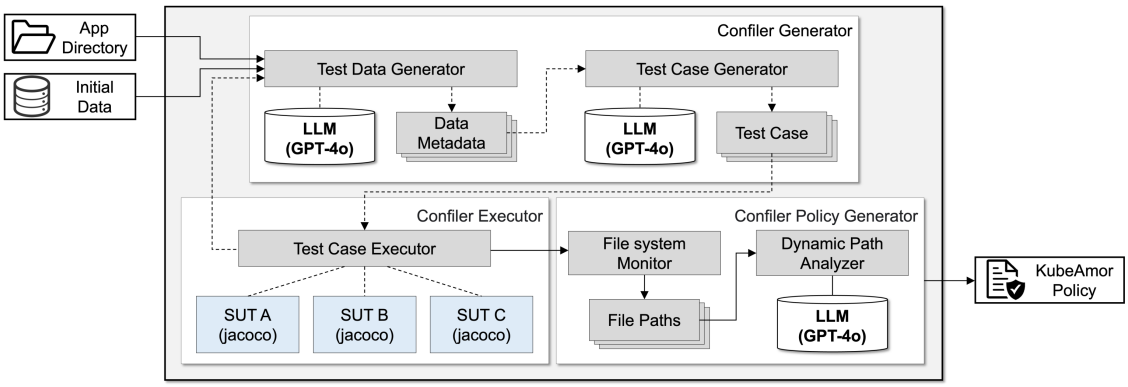

Figure 1. Confiler 아키텍처의 구성요소: Confiler Generator, Confiler Executor, Confiler Policy Generator

테이너 기반 시스템이 확산됨에 따라 파일 시스템 접근 제어는 보안의 핵심 과제로 자리 잡았다. 접근 제어는 블랙리스트 방식과 화이트리스트 방식으로 구분되며, KubeArmor는 주로 블랙리스트 기반 정책을 중심으로 CVE와 같은 공격에 대응할 수 있는 정책 예시를 제공한다.

2.2. Motivation

블랙리스트 기반 정책의 한계. 블랙리스트 기반 접근 제어 정책은 이미 식별된 위협이나 알려진 공격 패턴에 대해서는 효과적인 대응이 가능하나 알려지지 않은 위협, 즉 Zero-day 공격에 대해서는 대응이 불가능하다는 한계를 지닌다.

기존 테스트 케이스 생성 기법의 한계. 화이트리스트 정책을 생성하기 위해서는 충분한 커버리지를 갖춘 테스트 케이스가 필요하다. 그러나 현재 널리 사용되는 자동 테스트 케이스 생성 프레임워크인 Evomaster[2]는 실제 컨테이너를 테스트 대상으로 지정하기 위해 Software Under Test(SUT) Driver를 직접 작성해야 하며 이 과정에서 구성이 복잡하고 대상 애플리케이션의 전체 소스 코드를 이해하고 있어야 한다. 또한 내부적으로 랜덤 값을 이용해 파라미터를 생성하기 때문에 테스트 실행 시간이 길어야 더 높은 품질의 테스트 케이스를 생성할 수 있다. 이러한 구조적 특성으로 인해 서비스 구조가 빈번히 변경되는 마이크로서비스 환경에서는 적용하기 어렵다.

동적으로 결정되는 파일 경로 존재. 애플리케이션의 기능에 따라 파일 경로가 동적으로 생성되는 경우가 빈번하게 발생한다. 경로를 고려하지 않고 정적으로 정책을 생성하는 경우 실제 런타임 환경에서 정상적인 접근이 차단되어 컨테이너의 동작을 방해할 수 있다. 반면 과도하게 일반화된 경로 허용은 공격 표면의 확대로 이어질 수 있다.

III. System Design

4.1. Confiler Overview

본 논문에서는 §2.2절에서 제시한 세 가지의 도전 과제를 효과적으로 해결하기 위해 사용자의 적은 개입과 시간으로 정교한 화이트리스트를 자동으로 생성하는 기법을 제안한다.

4.2. Confiler Generator

Test Data Generator는 소스 코드에서 정적 분석을 통해 entry point 함수를 식별한다. 또한 파라미터 흐름을 분석해 data flow graph를 구축하고 사용자가 제공한 초기 데이터는 실제 데이터 베이스와 매핑되어 저장된다. 이후 entry point 및 관련 함수의 소스 코드, 매핑된 메타데이터를 바탕으로 분기 문맥을 이해하고 적절한 파라미터를 생성하기 위해 LLM(GPT-4o)에 질의한다. 이를 통해 다양한 분기 경로를 포괄하는 테스트 케이스 입력값을 생성하며, 생성된 입력값은 Test Case Generator로 전달된다. Test Case Generator는 API 유형과 메타데이터를 기반으로 LLM을 활용해 앞서 생성한 입력값으로부터 실행 가능한 테스트 케이스를 생성한다.

4.3. Confiler Executor

생성된 테스트 케이스는 Test Case Executor에 의해 테스트 컨테이너에 적용된다. 이후, JaCoCo를 활용하여 테스트 결과를 확인하고 오류가 발생하면 커버리지를 높이기 위해 Test Data Generator로 되돌아가 새로운 데이터를

	core-banking	fund-transfer	user	utility-payment
Evomaster	2/6	2/2	1/4	1/2
Confiler	**6/6**	2/2	**3/4**	**2/2**

Table 1. 테스트 케이스의 엔드포인트 호출 성공 여부 비교 (vs. Evomaster)

생성한다. 반면, 성공 시에는 입력값과 응답이 피드백되어 이후 테스트 생성에 반영된다.

4.4. Confiler Policy Generator

성공적으로 실행된 테스트 케이스는 Falco 기반 모니터링을 통해 실제 접근한 파일 경로를 수집한다. 이때 일부 경로는 런타임에서 동적으로 결정되므로, 시스템은 few-shot 학습을 기반으로한 LLM 질의를 통해 동적 요소를 식별하고 이를 와일드카드로 일반화한다. 최종적으로 식별된 경로는 KubeArmor 형식의 화이트리스트 정책으로 변환되어 적용된다.

IV. Implementation & Evaluation

5.1. Implementation

본 논문에서는 제안한 기술의 실현 가능성을 검증하기 위한 프로토타입을 구현하였다. 테스트 케이스의 생성을 위한 소스코드 분석은 Java 언어 기반의 Javaparser 패키지를 활용하였으며, Evomaster의 테스트 케이스 생성 시간은 1시간으로 설정하였다.

5.2. Evaluation

테스트 케이스 커버리지 비교. 본 논문은 테스트 케이스의 커버리지 측면에서 Confiler의 효과성을 평가하였다. 이를 위해 Internet Banking Concept Microservice GitHub 프로젝트 내의 네 가지 서비스를 대상으로 엔드포인트 커버리지 비교 실험을 수행하였다. Table 1에서 확인할 수 있듯이 core-banking-service의 경우 기존 도구인 Evomaster는 6개의 엔드포인트 중 2개의 엔드포인트를 커버하여 낮은 커버리지를 보인 반면, Confiler는 모든 엔드포인트를 테스트 실행에 성공시켜 높은 커버리지를 달성하였다. 이는 Confiler가 LLM 기반 테스트 케이스 생성을 통해 보다 많은 코드 분기를 커버할 수 있음을 보여준다.

동적 경로 일반화 효과성. 본 논문은 동적 경로 일반화 과정에서 LLM의 few-shot 학습 효과를 평가하였다. 이를 위해 실제 애플리케이션에서 생성된 4개의 동적 경로 데이터를 기반으로, instruction만 제공된 모델(A)과 few-shot

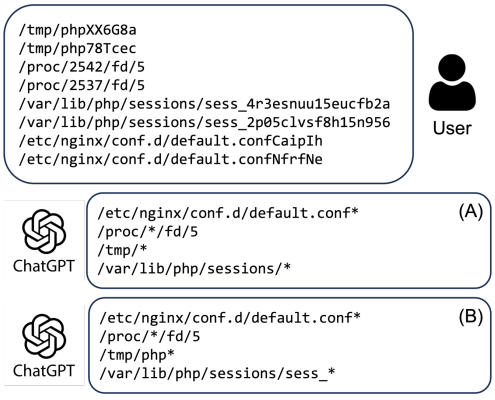

Figure 2. few-shot 학습의 유무에 따른 LLM의 일반화 답변 비교

예시를 포함한 모델(B)의 일반화 정확도를 비교하였다. Figure 2에서 보이듯, Figure 2(A)는 경로의 맥락을 이해하지 못하고 /tmp/, /var/lib, /php/sessions/ 등을 *로 단순 처리한 반면, Figure 2(B)는 /tmp/php*, /var/lib/php/sessions/sess_*처럼 의미 있는 패턴으로 정밀하게 일반화하였다. 이를 통해 few-shot 학습이 과도한 일반화를 줄이고, 경로 일반화의 정확도와 신뢰도를 높이는 데 효과적임을 확인하였다.

V. Conclusion

본 논문에서는 기존 블랙리스트 기반 접근 제어의 한계를 해결하기 위해 Confiler 프레임워크를 제안하였다. Confiler는 LLM을 활용한 테스트 케이스 자동 생성과 동적 경로 식별을 결합한 화이트리스트 정책 생성 기법을 제안하였다. 평가 결과, 기존 도구보다 높은 커버리지를 달성하며 제안 방식의 실효성을 입증하였다.

Acknowledgements

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2025-16069415).

[참고문헌]

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

[2] Arcuri, Andrea. "RESTful API automated test case generation with EvoMaster." ACM Transactions on Software Engineering and Methodology (TOSEM)28.1 (2019): 1-37.