LSM-BPF 기반 컨테이너 보안 정책 적용에 관한 성능 분석

김이수[1], 최승연[2], 남재현[3]
[1]단국대학교 모바일시스템공학과 학부생
[2]단국대학교 컴퓨터공학과 학부생
[3]단국대학교 컴퓨터공학과 교수

isukim@dankook.ac.kr, 32204601@dankook.ac.kr, namjh@dankook.ac.kr

An Analysis on Performance with LSM-BPF based Container Security Enforcement

Isu Kim[1], Seungyeon Choi[2], Jaehyun Nam[3]
[1]Dept. of Mobile Systems Engineering, Dankook University (Undergraduate Student)
[2]Dept. of Computer Engineering, Dankook University (Undergraduate Student)
[3]Dept. of Computer Engineering, Dankook University (Professor)

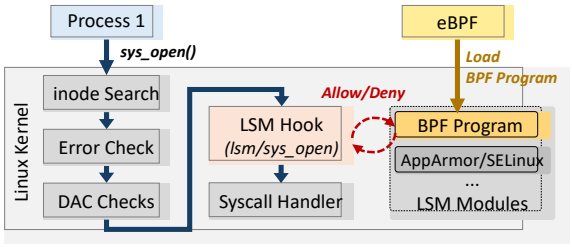
요 약

클라우드 환경에서 컨테이너 사용이 증가하면서 컨테이너 보안의 중요성이 부각되고 있다. 그리고, 이를 위해 다양한 리눅스 보안 프레임워크를 활용한 연구가 활발하게 진행되고 있으며, 특히 LSM-BPF 기술이 기존의 보안 프레임워크 한계를 극복할 수 있는 해법으로 주목받고 있다. 그러나 LSM-BPF 를 활용할 때 발생할 수 있는 시스템 성능 변화에 대한 연구는 아직 미흡한 상태이다. 따라서, 본 논문에서는 LSM-BPF 기반의 보안 솔루션을 모의 구현하고 이를 벤치마크하여 시스템 성능 변화를 측정 및 분석을 진행하였다. 컨테이너 환경에서의 실험 결과, 평균적으로 시스템 성능이 6.6% 감소하는 것을 확인하였다. 하지만, 보안 정책 개수가 증가하여도 추가적인 성능 저하는 발생하지 않음 역시 확인하였다.

1. 서론

최근 클라우드 환경에서는 개발의 신속화와 배포의 통일성이라는 장점을 가진 컨테이너 사용이 증가하고 있다. 그러나, 이에 따른 보안 문제 또한 심각해지고 있으며, 이는 컨테이너화된 애플리케이션이 기존의 애플리케이션과 보안 취약점 및 공격 경로를 공유하기 때문이다[1]. 결국, 공격자들은 탈취된 컨테이너 내부에서 추가적인 악성 프로그램을 다운로드하고 실행함으로써 클러스터 전반의 보안을 위협한다[2].

하지만, 최신 클라우드 환경에서는 기존의 애플리케이션보다 내부 동작을 관찰하기 어려운 컨테이너를 사용하기에 기존 보안 솔루션으로는 한계가 존재한다[3]. 따라서, Seccomp, AppArmor, SELinux 와 같은 리눅스 보안 프레임워크를 기반으로 컨테이너에 사용자 정의 보안 정책을 적용하는 연구가 활발하게 진행되고 있다[4]. 그러나, 이러한 프레임워크는 특정 운영체제에서만 동작하거나[5], 컨테이너 실행 후 제한적 정책 수정만 가능하다[6]는 한계점을 가진다.

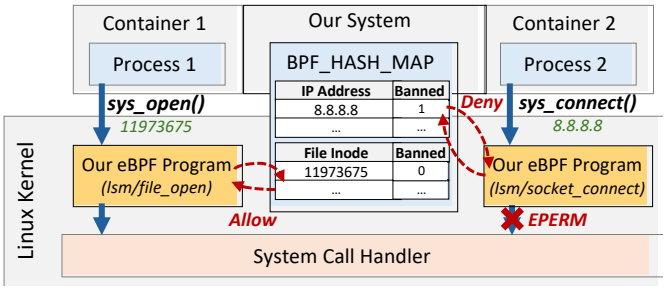


(그림 1) LSM-BPF 동작 구조

최근 주목받는 LSM-BPF (Linux Security Module Berkley Packet Filter)는 eBPF (Extended BPF)에서 제공하는 커널 기능을 활용하여 기존 보안 프레임워크의 문제점을 보완할 수 있는 기술이다[7]. 그림 1 과 같이 LSM-BPF 는 eBPF 프로그램을 임의의 LSM 훅에 부착하여 강제적 접근 제어(MAC)를 달성할 수 있다[8].

하지만, LSM-BPF 사용에 따른 문제점 중 성능 변화에 대한 연구는 충분히 이루어지지 않았다. 따라서, 본 논문에서는 LSM-BPF 기반 보안 솔루션을 모의 구현하고, 벤치마크를 통해 LSM-BPF 를 사용할 때의 시스템 성능 변화를 측정하고 분석한다.

2. 모의 LSM-BPF 솔루션 구현



(그림 2) 모의 LSM-BPF 솔루션 구조

구현된 모의 LSM-BPF 솔루션은 컨테이너 내에서 파일 접근(file_open)과 네트워크 연결(socket_connect)과 같은 시스템 동작을 제어한다. 이 솔루션은 그림 2 와 같이 사용자 공간의 사용자 보안 정책을 커널 공간의 eBPF 프로그램으로 전달하고 적용하기 위해 eBPF 의 다양한 자료구조 중 해시 맵을 활용한다.

3. LSM-BPF 성능 변화 측정

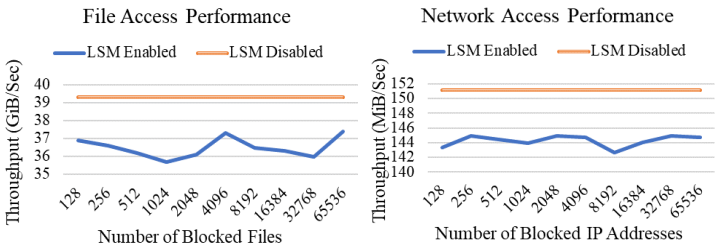
LSM-BPF 사용에 따른 파일 접근 및 네트워크 연결 성능 변화를 측정하기 위해, 표 1 과 같은 환경에서 구동되는 컨테이너에서 시스템 기능을 활용하는 벤치마크 도구인 Fio 와 wrk 를 사용하여 처리 지연 시간 및 처리량을 측정하였다. 표 2 는 모의 보안 솔루션 적용에 따른 컨테이너 성능 변화를 보여주며, 그림 3 은 차단된 파일 개수 및 IP 주소에 따른 처리량 변화를 나타낸다.

종류	항목
CPU	AMD Ryzen9 5900X (12C 24T)
RAM/SSD	64GB / Samsung 980 PRO
OS/SW	Ubuntu 22.04 (5.15.0-67), Docker 23.0.1

<표 1> 실험 환경

	파일 접근[1]		네트워크 연결[2]	
적용 유무	지연시간 (ns)	처리량 (GiB/s)	지연시간 (ns)	처리량 (MiB/s)
O	3013.84	36.5	67.515	144.241
X	2772.76	39.3	63.71	151.09

<표 2> 파일 접근 및 네트워크 연결 성능 변화



(그림 3) 파일 및 IP 주소 차단에 따른 성능 변화

[1] Fio 를 통한 1000 개의 파일 생성 후 임의 작성
[2] wrk 를 통한 컨테이너간 24 개의 동시다발적 통신

4. LSM-BPF 성능 변화 확인 및 원인 분석

실험을 통해, LSM-BPF 를 활용할 경우, 사용하지 않을 때보다 파일 접근은 약 7.1%, 네트워크 연결은 5.5%의 성능 저하가 발생하는 것을 확인하였다. 이러한 성능 저하는 eBPF 프로그램이 시스템 동작의 허가 여부를 결정하고 종료되기까지 해당 시스템 동작이 대기 상태에 머무르면서 발생하는 지연이 쌓여 발생하였다. 반면에, 차단된 파일이나 IP 주소의 개수가 증가해도 성능에 유의미한 변화가 없는 것으로 나타났다. 이는 eBPF 프로그램이 해시 맵을 이용해 차단 대상을 확인하기 때문에 차단 대상의 수가 늘어나더라도 시간 복잡도가 거의 일정하여 성능에 미치는 영향이 적기 때문이다.

5. 결론

본 논문에서는 모의 프로그램을 통하여 컨테이너 환경에서 동작하는 LSM-BPF 프로그램의 여부와 차단 대상의 개수에 따른 컨테이너의 성능 변화를 측정 및 분석하였다. 이를 통하여 커널 공간에서 동작하는 eBPF 프로그램의 시스템 행위 의사 결정 로직 자체는 성능 저하에 기여하나, 차단 대상의 개수는 해쉬 맵을 사용하는 경우 성능 변화에 큰 영향이 없음을 확인할 수 있었다. 향후 연구로는 LSM-BPF 의 eBPF 프로그램의 최적화에 대한 연구를 진행할 계획이다.

Acknowledgement

이 성과는 2024 년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (RS-2023-00212738)

참고문헌

[1] Murugiah Souppaya, et al., Application Container Security Guide. NIST. 2017.09.
[2] Sari Sultan, et al., (2019). "Container Security: Issues, Challenges, and the Road Ahead". IEEE Access, Vol 7, pp. 52976-52996.
[3] Vivek Vijay Sarkale, et al. "Secure Cloud Container: Runtime Behavior Monitoring using Most Privileged Container (MPC)" IEEE Computer Society, 2017.
[4] Yuqiong Sun, et al. "Security Namespace: Making Linux Security Frameworks Available to Containers." USENIX Security Symposium, USENIX, 2018.
[5] AppArmor, https://apparmor.net/
[6] "Seccomp BPF", https://www.kernel.org/doc/html/v4.18/userspace-api/seccomp_filter.html
[7] Sharaf, H., et al. "Extended Berkeley Packet Filter: An Application Perspective", In IEEE Access Vol. 10 (2022)
[8] "LSM BPF Programs", https://docs.kernel.org/bpf/prog_lsm.html