2024년 한국정보보호학회 하계학술대회

CISC-S'24

Conference on Information Security and Cryptography Summer 2024

일자 2024년 6월 20일(목) ~ 21일(금)

장소 세종시 정부세종컨벤션센터

주최 한국정보보호학회 Korea Institute of Information Security & Cryptology

주관 고려대학교 세종캠퍼스 KOREA UNIVERSITY SEJONG CAMPUS

후원 국가정보원 NATIONAL INTELLIGENCE SERVICE, 과학기술정보통신부, 행정안전부, 세종특별자치시, KISA 한국인터넷진흥원, ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute, NSR 국가보안기술연구소 National Security Research Institute, KISTI 한국과학기술정보연구원 Korea Institute of Science and Technology Information www.kisti.re.kr

한국정보보호학회
Korea Institute of Information Security & Cryptology

CATS : 마이크로서비스 프로세스 안티-템퍼링을 통한 공격 탐지 방안 연구

김상현*, 최상훈[1], 박기웅†

세종대학교 정보보호학과 (대학원생, 연구 교수[1], 교수†)

CATS: A Study on Attack Detection Methods Through Anti-Tampering in Microservice Processes

Sang-Hyeon Kim*, Sang-Hoon Choi[1], Ki-Woong Park†

*Dept. of Computer and Information Security, Sejong University

(Graduate student*, Research Professor[1], Professor†)

요 약

최근 인프라 시장은 온-프레미스에서 마이크로서비스 아키텍처를 채택한 클라우드 네이티브 환경으로 전환되고 있다. 이를 통해 세부적인 서비스 관리가 가능해졌지만, 보안 관리 비용이 증가했다. 본 논문에서는 초기 프로세스를 기록하고 변화 탐지 및 대응을 통해 보안을 유지하는 Container Anti-Tampering System(CATS) 프레임워크를 제안한다. CATS는 초기 프로세스를 캡처하고 변화 시 대응하여 무결성을 유지하며, 성능저하는 최소화된다. 실험 결과, CVE-2023-50164 환경에서 평균 887,004ns의 탐지 시간과 초당 요청 처리량에서 약 0.504%, 초당 응답 처리량에서 약 0.503%의 성능 오버헤드를 보였다. CATS 프레임워크는 컨테이너 보안의 효율성을 증대시키고, 서비스 무결성을 유지하는 데 기여할 것으로 기대된다.

I. 서론

최근 인프라 시장은 온-프레미스(On-premises) 환경에서 마이크로서비스 아키텍처(MicroService Architecture)를 채택한 클라우드 네이티브(Cloud Native) 환경으로 변화하고 있다[1, 2]. 마이크로서비스는 애플리케이션의 서비스를 더 세분화하여 컨테이너라고 불리는 기능 단위로 관리할 수 있도록 운용한다. 기능을 분리함으로써 서비스를 세부적으로 관리하는 측면에서는 이점이 있지만, 공격 범위가 늘어났고 보안 정책을 부여하고 관리하는데 비용이 증가했다고 볼 수 있다.

세분화된 서비스들은 대부분 필요한 명령어나 시스템콜이 한정되어 있다. 그러므로 보통 컨테이너마다 Seccomp와 같은 필터링 보안 도구를 사용하여 불필요한 명령어를 지정하여 보안을 유지한다. 하지만 컨테이너마다 각각 지정해야 하므로 휴먼에러에 노출되어 있고 적지 않은 시간과 노력이 소모된다. 만약 필터링 보안 도구의 잘못된 설정으로 악의적인 행위에 사용되는 명령어나 시스템콜이 허용되어도 서비스하는 동안 프로세스의 변화가 없는 컨테이너는 프로세스 변화를 탐지하는 것만으로도 악성 행위를 방지할 수 있다. 본 논문에서는 위와 같은 상황에서 초기의 프로세스를 기록하고 후에 프로세스의 변화를 탐지하고 대응 조치하는 Container Anti-Tampering System(CATS) 프

† 교신저자: 박기웅 (세종대학교 정보보호학과 교수)

본 논문은 과학기술정보통신부의 재원으로 정보통신기획평가원(IITP)의 정보통신방송기술 국제공동연구(Project No. RS-2022-00165794, 40%), 국방ICT융합연구(Project No. 2022-11220701, 20%), 실감콘텐츠핵심기술개발(ProjectNo. RS-2023-00228996, 10%), 정보통신방송혁신인재양성사업(Project No. 2021-0-01816, 10%) 및 한국연구재단(NRF) 중견후속연구사업(Project No. RS-2023-00208460, 20%)의 지원을 받아 수행된 연구임.

레임워크를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 컨테이너 보안 및 이상 징후 탐지 관련 연구를 서술하고, 3장에서는 CATS 프레임워크의 디자인 및 구현을 서술한다. 4장에서는 성능평가를 서술하고 5장에서는 결론 및 향후 연구 방향을 제시한다.

II. 관련 연구

본 장에서는 컨테이너 보안 관련 연구 및 컨테이너 이상 징후 탐지 관련 연구를 살펴본다.

2.1. 컨테이너 보안 관련 연구

20년 Nuno Lopes et al. 연구진은 컨테이너 보안을 개선하기 위한 옵션 중 하나인 Seccomp 프로파일의 문제점을 지적했다[3]. 프로파일을 자주 업데이트해야 하며, 업데이트할 내용을 정확하게 결정하는데 복잡하고 시간이 많이 소요되는 어려움이 있다고 주장했다. 이를 해소하기 위해 본 논문에서는 Seccomp 보안 기능을 CI/CD(Continuous Integration/Continuous Development) 파이프라인과 결합하여 프로파일을 최신 상태로 유지하는 자동화된 Seccomp 프로파일링 기법을 제안했고, 프로파일에 없는 Syscall이 필요한 몇 가지 취약성을 완화할 수 있다고 시사했다.

21년 Michael Reeves et al. 연구진이 발표한 연구 결과에 따르면, 컨테이너를 탈출하기 위해 이용하는 취약성에는 프로파일 오설정, 리눅스 커널 버그, 컨테이너 런타임 버그 세 가지 주요 유형을 정의했다[4]. 본 논문에서는 11개의 컨테이너 런타임과 59개의 취약점에 관한 보안 연구를 수행했다. 그 후 공개적으로 사용 가능한 PoC 취약점이 있는 28개의 CVE에 대한 7개의 클래스 분류법을 제시하여 컨테이너 탈출의 주요 원인이 컨테이너로 유출된 호스트 구성 요소임을 밝혔다. 또, 컨테이너 런타임에 사용자 네임스페이스 방어를 적용하면 9개의 탈출 취약점 중 7개를 방어할 수 있다고 증명하며 컨테이너 탈출을 효과적으로 방지할 수 있다고 주장했다.

2.2. 컨테이너 이상 징후 탐지 관련 연구

21년 Zhaofeng Yu et al. 연구진은 기존 컨테이너 내 보안 도구가 컨테이너 내 악의적인 프로세스에 의해 탐지되거나 공격 될 수 있으며 권한 확장 공격이 수행될 수 있다고 주장했다[5]. 이 때문에 외부에서 컨테이너를 모니터링하는 것이 더 안전하다고 판단했다. 본 연구진은 이러한 보안 문제를 해결하기 위해 가상 머신 인트로스펙션 기술을 기반으로 컨테이너를 모니터링하는 외부 모니터링 접근 방식을 제안했다. 제안한 아키텍처에서 대상 컨테이너는 더 강력한 격리를 위해 가상 머신에서 실행되고 있고 보안 도구는 하이퍼바이저에서 작동한다. 모니터링 프로그램은 가상 머신 외부에서 실행되며 가상 머신에서 네임스페이스와 컨테이너 프로세스를 자동으로 식별하고 서로 다른 컨테이너의 프로세스에 다른 보안 분석을 수행한다. 분석 결과에 따라 실시간으로 대응할 수 있을 뿐만 아니라 컨테이너 탈출 행동도 모니터링할 수 있다고 주장했다.

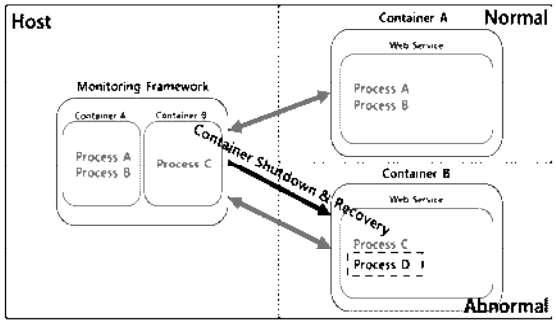
22년 Asbat El khairi et al. 연구진은 최첨단 이상 징후 기반 호스트 침입 탐지 시스템(HIDS)의 컨테이너화 환경에서의 한계점을 지적했다[6]. 위 시스템은 컨테이너 런타임 보안을 강화할 수 있지만, 컨테이너화 환경의 특성을 처리하도록 설계되지 않아 컨테이너의 확장성과 이상 징후의 다양성에 효과적으로 대처할 수 없다고 주장했다. 이러한 문제를 해결하기 위해 본 연구진은 시스템콜을 사용하여 컨테이너를 모니터링하기 위한 이상 징후 기반 침입 탐지 기술을 제시했다. 기존 솔루션과 달리 그래프 기반 모델을 활용하여 컨텍스트 내에서 서로 다른 Syscall 속성을 분석하여 이상 징후로 나타나는 고유 활동을 파악할 수 있고 이전에는 볼 수 없었던 공격에 대해 컨테이너를 효율적으로 모니터링할 수 있으며 포렌식, 트리아징 또는 사고 대응 등에 활용할 수 있다고 주장했다.

III. 디자인 및 구현

본 장에서는 우리가 제안하는 CATS 프레임워크를 설계하고 구현하는 과정을 설명한다.

3.1. CATS 프레임워크 설계

본 연구에서 제안한 모니터링 프레임워크는 불변한 프로세스로 서비스하는 컨테이너들의 초기 프로세스를 캡처하고 프로세스를 감시하고 있다가 프로세스의 변화가 감지되면 대응 조치하는 방식으로 동작 흐름은 (그림 1)과 같다.



(그림 1) 프레임워크 동작 흐름

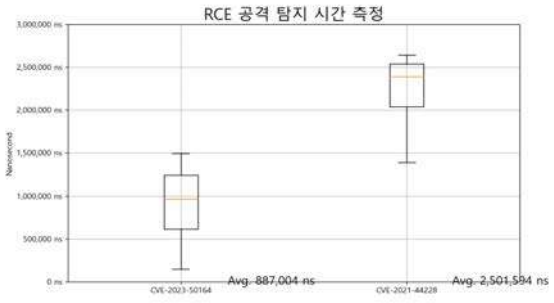
3.2. CATS 프레임워크 구현

CATS의 동작 과정은 다음과 같다. CATS 프레임워크는 호스트 운영체제에서 Docker로 인해 구동되는 컨테이너의 Cgroups 정보를 추적한다. Cgroups 분석이 완료된 이후 Cgroups 별 proc 파일 시스템에 있는 프로세스 정보를 기반으로 화이트리스트를 생성한다. 이후, proc 위변조 탐지를 위해 센싱을 수행한다. 센싱과정에서 화이트리스트에 등록되지 않은 프로세스가 proc 파일 시스템에서 감지되면 정상적이지 않은 RCE(Remote Code Execution) 공격이 발생한 것으로 판단한다. 프로세스 위변조가 탐지되면 해당 컨테이너를 같은 이미지의 컨테이너로 교체한다. 이를 수행함으로써 변조되지 않은 무결한 컨테이너 서비스를 제공할 수 있다.

IV. 성능평가

본 장에서는 제안하는 CATS 프레임워크의 실용성을 검증하기 위해 성능평가를 수행하고 그 결과를 서술한다.

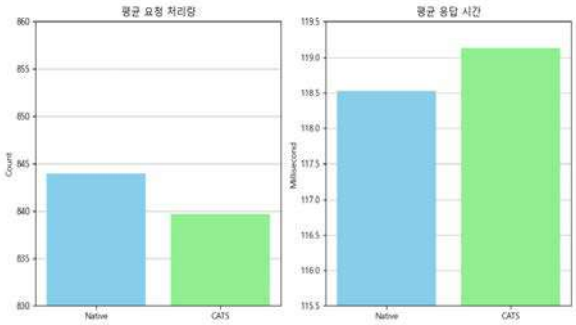
4.1. 공격 탐지 성능평가



(그림 2) RCE 공격 탐지 시간

우리는 컨테이너 오염을 탐지하기 위해 CVE-2023-50164 취약점을 가진 컨테이너 서비스 환경과 CVE-2021-44228 취약점을 가진 컨테이너 서비스를 구축했다[7, 8]. CVE-2023-50164 취약점이 있는 서비스에서는 Firefox를 사용하여 웹셸 업로드를 통해 RCE 공격을 수행하고, 공격을 탐지하기까지 걸리는 시간을 측정하였다. CVE-2021-44228 취약점이 있는 서비스에서는 Firefox를 사용하여 악성 페이로드를 입력해 RCE 공격을 수행하고, 공격을 탐지하기까지 걸리는 시간을 측정하였다. 실험 결과는 (그림 2)와 같다. 우리의 실험 결과, 프로세스 변조를 탐지하는데 CVE-2023-50164 환경에서 평균 887,004ns, CVE-2021-44228 환경에서 평균 2,501,594ns가 소요되는 것을 확인했다. 즉, 서비스의 취약점에 의해 공격을 당하더라도 매우 빠르게 감지할 수 있어 서비스 탈취 및 정보 유출을 예방할 수 있음을 보여준다.

4.2. 프로세스 필터링으로 인한 서비스 저하 성능평가



(그림 3) 평균 요청 처리량 및 응답시간

CATS 프레임워크는 컨테이너별로 생성되는 Cgroups의 proc 파일시스템을 센싱하여 프로세스 변화 정보를 탐지한다. 따라서 모니터링으로 인한 서비스 성능저하가 발생할 수 있다. 우리는 익스플로잇을 탐지했던 환경인 Apache Struts2(CVE-2023-50164)에서 프로세스 센싱으로 인한 성능저하를 ApacheBench를 통해 평가하였다. 실험 결과는 (그림 3)과 같다. 모니터링을 수행하지 않는 서비스의 경우 초당 요청 처리량이 평균 843.981이었으며, 우리의 프레임워크가 적용된 서비스의 경우 초당 처리량이 839.726이었다. 추가적으로, 프로세스 모니터링이 되지 않는 서비스에서는 요청에 따른 응답시간이 평균 118.5268ms가 소요되었으며, 우리의 프레임워크가 적용된 환경은 평균 119.1233ms를 보였다. 결과적으로 초당 요청 처리량은 약 0.504%, 응답시간의 경우 약 0.503%

의 오버헤드가 발생하였다. 이와 같은 결과는 CATS로 인해 서비스의 처리 성능저하가 거의 발생하지 않는다는 것을 보여준다.

V. 결론 및 향후 연구

본 논문은 프로세스 생성, 변경, 삭제가 불필요한 컨테이너를 대상으로 초기의 프로세스 상황을 기록한 후 프로세스의 변화를 탐지하여 대응 조치하는 CATS 프레임워크를 제안하였다. 제안한 프레임워크의 실용성을 검증하기 위해 CVE-2023-50164 취약점을 적용한 환경에서 익스플로잇 탐지 성능평가와 프로세스 필터링으로 인한 서비스 저하 성능평가를 수행한 결과 평균 887,004ns의 탐지 시간과 초당 요청 처리량 및 응답시간이 각각 0.504%, 0.503%의 오버헤드가 발생했다. 위 결과로 보아 최소한의 오버헤드로 프로세스의 변조를 탐지 가능하다고 볼 수 있다.

향후 연구에서는 웹 서비스뿐만 아니라 적용할 수 있는 다른 유형의 서비스를 탐색하고 공격 패턴을 학습하여 같은 공격에 대비하는 메커니즘을 연구하고자 한다.

[참고문헌]

[1] Koshy, Jayden et al. From On-Premises to Cloud: Crafting Your Pathway for Migration Success. preprints preprint preprints:202311.0841.v1. 2023.

[2] V. Singh and S. K. Peddoju, Container-based microservice architecture for cloud applications, 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 847-852, May, 2017.

[3] Nuno Lopes et al. Container Hardening Through Automated Seccomp Profiling. 2020 6th International Workshop on Container Technologies and Container Clouds (WOC'20). Association for Computing Machinery, pp. 31 - 36, 2020.

[4] M. Reeves et al. Towards Improving Container Security by Preventing Runtime Escapes. 2021 IEEE Secure Development Conference (SecDev), pp. 38-46, 2021.

[5] Yu, Zhaofeng et al. A Container-Oriented Virtual-Machine-Introspection-Based Security Monitor to Secure Containers in Cloud Computing. (eds) Artificial Intelligence and Security. ICAIS 2021. Lecture Notes in Computer Science(), vol 12737. Springer, Cham., pp. 102-111. 2021.

[6] Asbat El Khairi et al. Contextualizing System Calls in Containers for Anomaly-Based Intrusion Detection. 2022 on Cloud Computing Security Workshop (CCSW'22). Association for Computing Machinery, pp. 9 - 21, 2022.

[7] CVE-2023-50164, http://github.com/Trackflaw/CVE-2023-50164-ApacheStruts2-Docker

[8] CVE-2021-44228, https://github.com/kozmer/log4j-shell-poc