Docker 컨테이너 취약점 사례 분석: 컨테이너 탈출 취약점

곽도원[1], 최형기[2]
[1]성균관대학교 소프트웨어학과 학부생
[2]성균관대학교 소프트웨어학과 교수

hegnut5859@gmail.com, meosery@skku.edu

Analysis of Docker Container Vulnerability Cases: Container Escape Vulnerability

Do-Won Kwak[1], Hyung-Kee Choi[1]
[1]Dept. of Computer Science and Engineering, Sungkyunkwan University

요 약

컨테이너 기술은 Virtual Machine(이하 VM) 보다 애플리케이션의 경량화와 배포 간소화에 유리해 클라우드 환경에서 널리 사용되고 있다. 그러나 VM에 비해 보안 측면에서 더 많은 위험에 노출될 수 있으며, 특히 컨테이너 탈출(Container Escape) 취약점은 큰 위협이 된다. 이는 격리된 컨테이너가 호스트 시스템에 접근하는 심각한 보안 문제로, 본 논문에서는 주요 컨테이너 탈출 취약점을 이용, 공격을 실제로 진행하면서 원인을 파악하여 Docker 기반 환경의 보안 강화를 위한 실질적인 참고자료를 제공하고자 한다.

1. 서론

컨테이너는 경량화된 애플리케이션 실행 환경을 제공하며, VM보다 빠른 속도와 높은 효율성으로 널리 사용되고 있다. 그러나 호스트 OS를 공유하는 특성 때문에 보안 측면에서는 상대적으로 취약하다.

특히 컨테이너 탈출은 격리 기능이 무력화되어 호스트 권한을 탈취할 수 있는 심각한 위협이다. 이를 방지하기 위해 seccomp, AppArmor, namespace 등 다양한 보안 메커니즘과 gVisor, Kata Containers와 같은 보안 중심의 런타임이 제안되고 있다.

본 논문에서는 대표적인 컨테이너 탈출 취약점을 실험적으로 재현하고, 각 취약점의 공격 시점, 원인, 그리고 대응 방안을 분석하고자 한다.

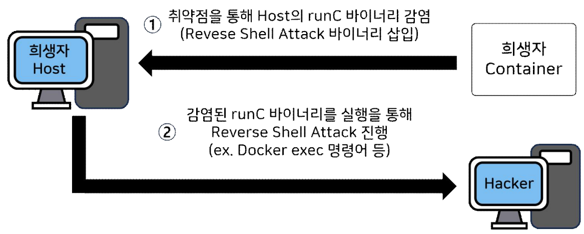
2. 재현 시스템 구성

공격자 VM과 피해자 VM은 브릿지 네트워크로 연결된 Ubuntu 환경으로 설정하고, 각 취약점에 맞는 Docker 및 커널 버전으로 환경을 구성하였다.

3. CVE-2019-5736 취약점

CVE-2019-5736은 Docker 18.06.2 이전 버전에서 발견된 취약점[1]으로, runC 바이너리가 실행 중에도 /proc/self/exe를 통해 덮어쓰기 가능한 구조로 인해 발생한다. 공격자는 악성 Docker 이미지로 runC를 덮어쓰고 reverse shell을 실행한다.

해당 취약점은 runC를 읽기 전용으로 설정하여 보안을 강화함으로써 해결할 수 있다.

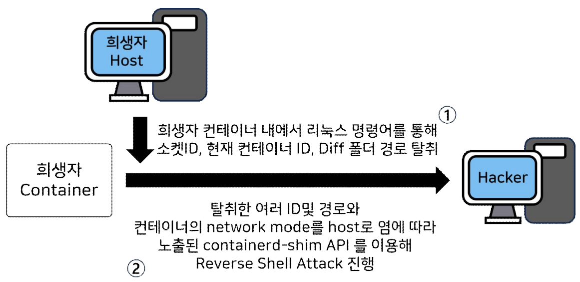


(그림 1) CVE-2019-5736 취약점 진행 과정

4. CVE-2020-15257 취약점

CVE-2020-15257은 containerd 1.3.8 이전 버전에서 Host 네트워크 모드 컨테이너가 Abstract Unix Domain Socket에 접근 가능해 권한 탈취가 가능한 취약점이다. 공격자는 Host 모드 컨테이너로 소켓에 접근해 API 호출을 통해 권한을 얻는다.

해당 취약점은 AppArmor로 소켓 접근을 차단하거나 최신 버전에서 업데이트 하여 File-Based Socket으로 변경해 해결한다.

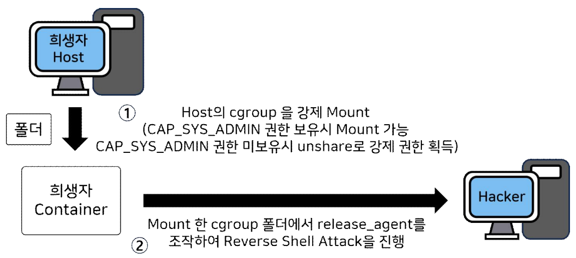


(그림 2) CVE-2020-15257 취약점 진행 과정

5. CVE-2022-0492 취약점

CVE-2022-0492는 Linux 커널 5.17 rc3 이전 버전에서 발견된 취약점[3]으로, cgroup의 release_agent 기능이 container 내에서 root 권한으로 실행 가능해 발생한다. 공격자는 권한 없이 release_agent를 조작하여 권한을 상승시키고 reverse shell을 실행한다.

해당 취약점은 커널 패치를 통해 해결하거나 사용자 네임스페이스 비활성화를 통해 대응 가능하다.

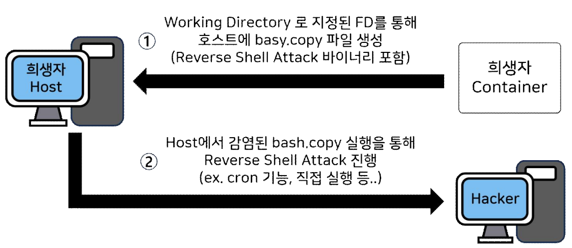


(그림 3) CVE-2022-0492 취약점 진행 과정

6. CVE-2024-21626 취약점

CVE-2024-21626은 runC 1.1.11 이전 버전에서 발생한 취약점[4]으로, runC가 열려 있는 디렉토리 FD를 Working Directory로 설정 가능해 발생한다. 공격자는 FD를 열고 악성 바이너리를 복사한 뒤 사용자가 이를 실행하도록 유도하여 탈출을 시도한다.

해당 취약점은 도커파일에서 Working Directory 설정을 제한하여 해결할 수 있다.



(그림 4) CVE-2024-21626 취약점 진행 과정

7. 각 취약점 공통점 및 차이점 분석

모든 취약점은 특정 런타임 또는 커널 컴포넌트의 구조적 결함에서 비롯되었으며 대부분 권한 상승 후 reverse shell로 연결된다. 또한 각 취약점들은 공격 시점과 원인에서 차이를 보인다는 것을 확인하였다.

<표 1> 각 취약점 공통점 및 차이점 분석

	CVE-2019-5736	CVE-2020-15257	CVE-2022-0492	CVE-2024-21626
컨테이너 탈출	ㅇ	ㅇ	ㅇ	ㅇ
취약점 공격 시점	희생자 결정	공격자 결정	희생자 결정	공격자 결정
취약점 발생 원인	runC	containerd	linux kernel	runC

8. 결론

이번 연구를 통해 보안 패치가 지속적으로 이루어지고 있음에도 컨테이너 탈출 취약점은 여전히 매년 발생하고 있음을 알게 되었다. 특히 해당 취약점들은 단순한 코드 결함이 아닌, 컨테이너 구조적 설계 한계에서 비롯된 경우가 많다.

따라서 단순 패치만으로는 대응하기에 부족하며, AppArmor, seccomp 등 보안 기능의 종합적 적용이 필요하다.

본 논문은 주요 탈출 취약점의 재현과 구조적 분석을 통해, 향후 보안 정책 수립 및 시스템 설계 시 실질적인 참고자료로 활용될 수 있을 것이다.

Acknowledgement

본 논문은 현대차 정몽구 재단 장학생으로서 지원을 받아 수행된 연구입니다

참고문헌

[1] National Institute of Standards and Technology. (2019). CVE-2019-5736: Vulnerability Summary for CVE-2019-5736. NVD

[2] National Institute of Standards and Technology. (2020). CVE-2020-15257: Vulnerability Summary for CVE-2020-15257. NVD

[3] National Institute of Standards and Technology. (2022). CVE-2022-0492: Vulnerability Summary for CVE-2022-0492. NVD

[4] National Institute of Standards and Technology. (2024). CVE-2024-21626: Vulnerability Summary for CVE-2024-21626. NVD