악성 도커 이미지 탐지 시스템 구축

유재겸[1]*, 김우종[1], 남서현[1], 서민재[1], 이다영[1], 이유경[1], 장혜선[1], 이병천[1]
중부대학교 정보보호학과[1]

Building a Malicious Docker Image Detection System

JaeKyeom Yoo[1]*, WooJong Kim[1], SeoHyeon Nam[1], MinJae Seo[1], Dayoung Lee[1], Yukyeong Lee[1], Hyesun Jang[1], Byoungcheon Lee[1]

요약 : 오늘날 Devops 및 클라우드 환경의 증가로 인해, 도커(Docker) 사용 빈도가 높아지고 있으며, 이로 인해 도커는 공격자들의 주요 대상이 되고 있다. 본 연구에서는 악성 도커 이미지에 대한 효과적인 대응을 위하여 정적 분석, 동적 분석, 결과 보고서 제공을 포함한 3단계로 나누어진 탐지 시스템을 구축하였다.

Key Words : Docker, Sandbox, Trivy, CVE, Static analysis, Dynamic analysis

1. 서 론

도커(Docker)는 리눅스 컨테이너를 기반으로 하는 오픈소스 가상화 플랫폼이다. 도커 컨테이너를 이용해 누구나 손쉽게 개발 및 테스트환경을 구축할 수 있어, 많은 개인 및 기업이 도커를 활용하고 있다. 도커 허브는(Docker Hub)는 가장 인기 있는 무료 컨테이너 이미지 원격 저장소이다. 배포된 이미지를 활용하기 위해서, 도커 허브를 통해 이미지를 다운받는다.

그러나 도커 허브에 있는 다수의 이미지들에서 취약점이 발견되고 있다. 이미 알려진 취약점을 내포하고 있거나, 고의적으로 악성 코드를 삽입한 이미지들도 있었다[1]. 고의적으로 삽입한 악성코드에는 악성 스크립트, 암호화폐 채굴 코드 등이 있었다.

본 연구에서는 도커 이미지에 대한 정적 분석, 동적 분석을 통해 해당 이미지가 가지고 있는 취약성을 알아내고, 분석 결과를 바탕으로 한 결과 보고서를 제공하여 악성 도커 이미지에 대한 대응에 기여하고자 한다.

2. 본 론

2.1 시스템 설계

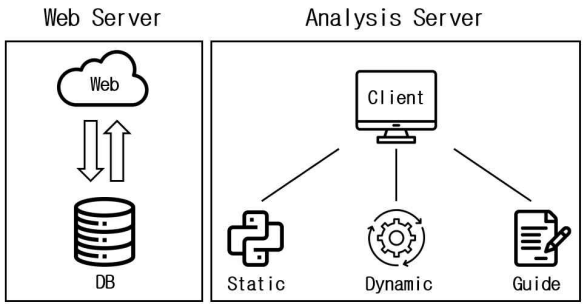

Fig. 1. System Configuration

탐지 시스템의 구성도는 위와 같다. 웹사이트를 통해 사용자가 분석을 원하는 이미지명을 입력하면, 해당 이미지에 대한 정적·동적 분석을 수행한다. 분석이 종료되면 분석 결과를 바탕으로 결과 보고서를 제공한다.

2.2 정적 분석

정적분석에서는 이미지에 대한 CVE 진단에 초점을 맞추었다. 오픈소스 취약점 스캐너인 트리비(Trivy)를 이용하여 다음과 같은 정적분석 결과를 제공한다.

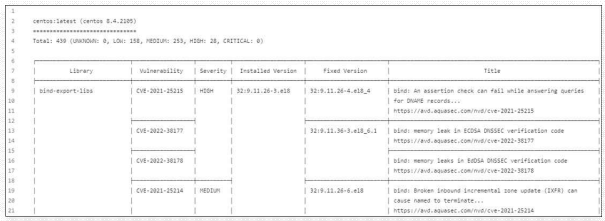

Fig. 2. Static Analysis Result

정적분석을 통하여 도커 이미지에서 발견된 CVE정보를 얻을 수 있다.

2.3 샌드박스 구축

분석해야 하는 이미지를 직접 실행하게 되면, 호스트 PC가 직접 감염될 우려가 있다. 이를 위해 동적 분석을 위한 샌드박스를 구축하였다.

Table 1. Sandbox Action Items

순번	수행목록
1	가상환경 구축
2	동적 분석 코드 전달
3	분석결과 반환

샌드박스에서는 분석 PC의 감염을 막기 위해서 분리된 가상 환경을 구축하였다. 가상환경에서 분석해야 할 도커 이미지를 실행한 후에, 동적 분석 코드를 전달한다. 이후 분석 결과를 DB에 저장한다.

2.4 동적 분석

동적 분석에서는 이미지 배포자가 고의로 삽입한 악성코드를 검사하는 데 초점을 맞추었다. 샌드박스를 통해 도커 컨테이너를 실행한 후, 아래와 같은 항목들을 탐지한다.

Table 2. Dynamic Analysis Detection Category

순번	탐지항목
1	관리자 권한 체크
2	네트워크 검사
3	열린 프로세스 검사
4	자동 실행 파일 검사

첫 번째로는 도커 이미지 관리자 권한 실행 여부를 검사한다. 도커 이미지 내에서 루트 권한으로 악의적인 프로세스를 실행할 경우, 호스트에 대한 관리자 권한을 탈취당할 우려가 있다(Docker Container Escape). 이와 같은 이유로 도커 이미지가 관리자 권한으로 실행되는지 검사하였다.

두 번째로는 네트워크 검사이다. 도커 이미지 내 암호화폐 채굴 또는 DOS(서비스 거부) 공격을 유발하는 코드가 숨겨져 있을 경우, 특정 IP와 통신을 하게 된다. netstat 명령어를 통해 현재 실행 중인 컨테이너가 외부와 통신하고 있는지 확인하고, 통신 IP를 추출하여 바이러스 토탈(virus total) API를 이용해 검사를 진행했다.

세 번째로는 열린 실행 프로세스 검사이다. 비 인가된 포트 및 프로세스는 시스템에 대한 악의적인 접근을 허용하게 한다. 이를 방지하기 위해 해당 컨테이너에서 열려있는 TCP 프로세스와 네트워크 연결에 대한 정보를 추출한다.

마지막으로 자동 실행파일을 검사한다. 악성 도커 이미지는 ENTRYPOINT를 사용해 컨테이너 내에 악성파일을 실행시키기도 한다. 우선 도커 히스토리 명령어를 통해, 이미지의 레이어 정보를 확인한다. 이미지 레이어의 ENTRYPOINT 구문에서 자동 실행 파일을 확인한다. 이후 해당 파일을 추출하여 컨테이너 외부로 복사하고, 바이러스 토탈 API를 통해 검사를 진행한다.

2.5 컨설팅 보고서 제공

Table 3. Vulnerability Detection Category

구분	진단항목
CVE 진단	오래된 소프트웨어 구성
	안전하지 않은 코딩
	네트워크 보안
	잘못 구성된 설정
	약한 인증 및 액세스 제어
권한 설정	관리자 계정 검사 : 탐지 / 미탐지
접근 통제	외부 아이피 통신 : 탐지 / 미탐지
	악성 아이피 분류 : 분류 / 미분류
	TCP 통신 프로세스 : 탐지 / 미탐지
실행파일 탐지	자동 실행 파일 검사 : 탐지 / 미탐지
	악성코드 분류 : 분류 / 미분류

컨설팅 전반적인 개요는 환경분석, 위험분석, 보안대책 순서대로 진행된다. 환경분석 단계에서는 요구사항을 정의하고, 현황 분석을 진행하여 진단 범위를 확정한다. 이를 통해 컨설팅 목표를 수립하였다.

본 연구에서는 도커 이미지 내에서 오래된 소프트웨어 라이브러리를 사용, 패치되지 않은 취약점 내포 혹은 악성코드를 포함하여 배포하는 행위를 탐지하고자 했다. 국내외 기술 자료를 바탕으로 위와 같이 점검 기준을 만들어, 진단 범위를 확정했다.

위험분석에서는 환경분석 단계를 통해 도출된 이행과제들에 대한 자산, 위협, 취약점 측면의 정도를 식별하고, 이를 통해 아래와 같은 기준으로 위험도를 산정하였다.

Table 4. Detection Vulnerability Risk Selection

위험도	내용
상	시스템·서비스의 가용성, 기밀성, 무결성 훼손
중	노출된 정보를 통해 추가 정보 유출 발생 우려
하	타 취약점과 연계할 수 있는 위협 내재

보안 대책에서는 사용자에게 결과보고서 및 조치 가이드를 제공하였다. 도커 이미지 취약점 분석 평가 항목과 그에 따른 위험도를 표시한 체크리스트, 분석 항목별 진단 방법과 조치 방법을 포함하여 제작하였다.

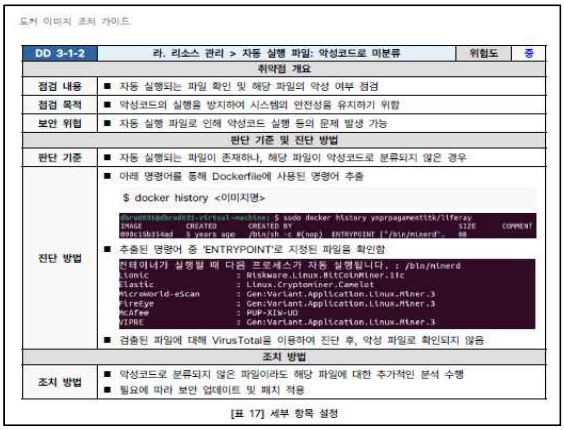

Fig. 3. Measures Guide

3. 결론

본 논문에서는 정적 및 동적 분석을 통해 도커 이미지의 취약성을 식별하고 보안성 향상을 위한 체크리스트를 제작하였다. 이를 통해 추후 악성 도커 이미지에 대한 효과적인 대응에 도움이 되기를 바란다.

참고문헌

[1] Myoungsung You, Jaehan Kim, and Seungwon Shin. "Revisiting Security Landscape of Docker Hub Container Images." The Journal of Korean Institute of Communications and Information Sciences 47, no. 8 (2022): 1231-1243, 10.7840/kics.2022.47.8.1231