Docker Malware Detection

팀 명 : 야자나무

지 도 교 수 : 이병천 교수님

팀 장 : 유재겸

팀 원 : 김우종

남서현

서민재

이다영

이유경

장혜선

2023. 11.

중부대학교 정보보호학과

목 차

1. 서 론

1.1 연구 배경 및 필요성

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 사용하기 위해서, 도커 허브를 통해 이미지를 다운받는다. 논문[컨테이너 이미지 보안성 분석에 관한 연구]에 따르면 도커 허브의 공식 이미지를 포함한 다수의 컨테이너 이미지에 다양한 취약점이 존재하며, 취약점에 대한 패치가 이미지에 적용될 때까지 상당한 시간이 소요되는 것으로 나타났다. 이에 대한 위협은 모두 해당 컨테이너를 내려받아 사용하는 사용자에게 전가된다. 도커 이미지 정적분석 및 샌드박스 기반 동적분석을 통해 도커 이미지 내 악성 행위 탐지율을 높이는 방안을 모색하고자 해당 연구를 기획하게 되었다.

1.2 연구 목적 및 주제선정

도커 이미지 분석을 통해 이미지 내 취약성 및 악성 행위를 파악하고, 분석 결과를 활용할 수 있도록 도커 이미지 분석 웹 서비스를 제작하고자 한다. 사용자가 웹 서비스 검색창에 특정 도커 이미지를 검색하면 해당 도커 이미지를 대상으로 정적· 동적 분석을 진행하여 결과 보고서 및 조치 가이드를 제공하도록 구현하는 것이 목표이다. 사용자가 직접 검사를 진행해야 하는 기존 도커 이미지 스캐너와 달리 자동 스캔 기능을 구현하여 보다 편리하게 이용할 수 있게 하고자 한다. 또한, 정적·동적 분석을 통해 정확한 탐지 결과를 내어 악성 도커 이미지에 미리 대비하고, 분석 결과에 따른 대응방안을 제시하여 2차 피해를 예방하고자 한다.

2. 관련연구

2.1 Docker

Docker는 리눅스 컨테이너에서 리눅스 애플리케이션을 프로세스 격리 기술을 사용하여 더 쉽게 컨테이너로 실행하고 관리할 수 있게 해주는 오픈소스 플랫폼이다. 격리된 공간에 필요한 라이브러리, 실행 파일만 담아놓고 사용하기 때문에 특정 환경에 구애받지 않는다는 장점이 있다. 도커는 최신 애플리케이션을 구축하고 배포하는데 이점을 제공하며, 실무에서 사용하는 서버 템플릿 도구 중 높은 비율을 차지하고 있다.

2.1.1 Dockerfile

Docker는 기본적으로 이미지가 있어야 컨테이너를 생성하고 동작시킬 수 있다. Dockerfile은 필요한 최소한의 패키지를 설치하고 동작하기 위한 자신만의 설정을 담은 파일이고, 이 파일로 이미지를 생성(빌드)하게 된다. 패키지 설치, 환경 변수 변경, 설정 파일 변경 등 다양한 작업을 하나하나 컨테이너를 만들고 설정을 적용할 필요 없이 Dockerfile을 사용하여 적용할 수 있고, 작업자의 실수로 인한 설정 누락 예방 등의 장점이 있다.

2.1.2 Docker image

Docker image는 애플리케이션 실행에 필요한 독립적인 환경을 포함하며, 런타임 환경을 위한 일종의 템플릿이다. 도커 이미지는 소스 코드, 라이브러리, 종속성, 도구 및 응용 프로그램을 실행하는데 필요한 기타 파일을 포함하는 불변 파일이다. 이미지는 읽기 전용이므로 스냅샷이라고도 하며, 특정 시점의 애플리케이션과 가상 환경을 나타낸다. 도커 이미지는 여러 개의 읽기 전용(read only) 레이어로 구성된다.

2.1.3 Docker container

Docker container는 도커 이미지를 실행한 상태로, 응용프로그램의 종속성과 함께 응용프로그램 자체를 패키징 또는 캡슐화하여 격리된 공간에서 프로세스를 동작시키는 기술이다.

2.2. Trivy

Trivy는 컨테이너 및 아티팩트에 대한 취약성 및 잘못된 구성정보에 대한 검사 도구이다. OS 패키지 (Alpine, RHEL, CentOS 등) 및 언어별 패키지 (Bundler, Composer, npm, yarn 등)의 취약점을 감지한다.

2.3 Sandbox

Sandbox는 외부로부터 들어온 프로그램이 보호된 영역에서 동작해 시스템이 부정하게 조작되는 것을 막는 보안 형태이다. 호스트 머신이나 운영 체제에 손상을 입히지 않고 확인되지 않거나 신뢰할 수 없는 서드파티, 공급자, 사용자, 웹사이트로부터 잠재적으로 테스트되지 않거나 신뢰하지 못하는 프로그램이나 코드를 실행하기 위해 사용된다

3. 본 론

3.1 웹 서비스 주소

http://dockerdetect.site

3.2 개발 환경

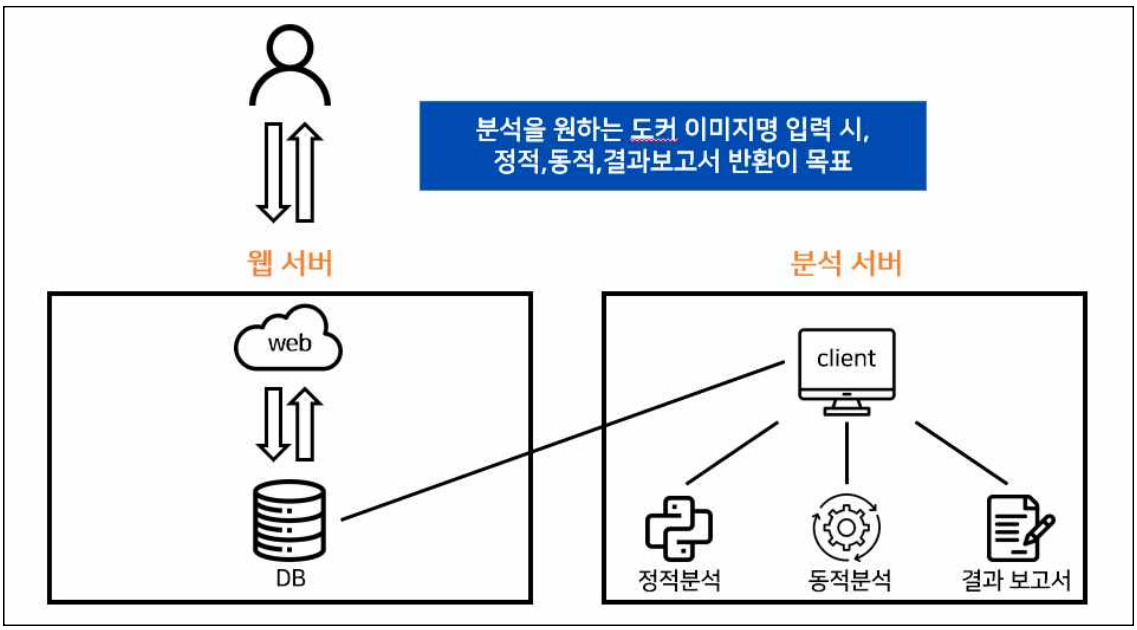
구분	항목
웹 서버(AWS), 분석 서버(Local)	Ubuntu 20.04
웹 개발	NodeJS, Express
DB	MongoDB
분석 자동화	Python

[표 1 서비스 개발 환경]

3.3 서비스 구성

서비스의 구성 및 흐름은 다음과 같다.

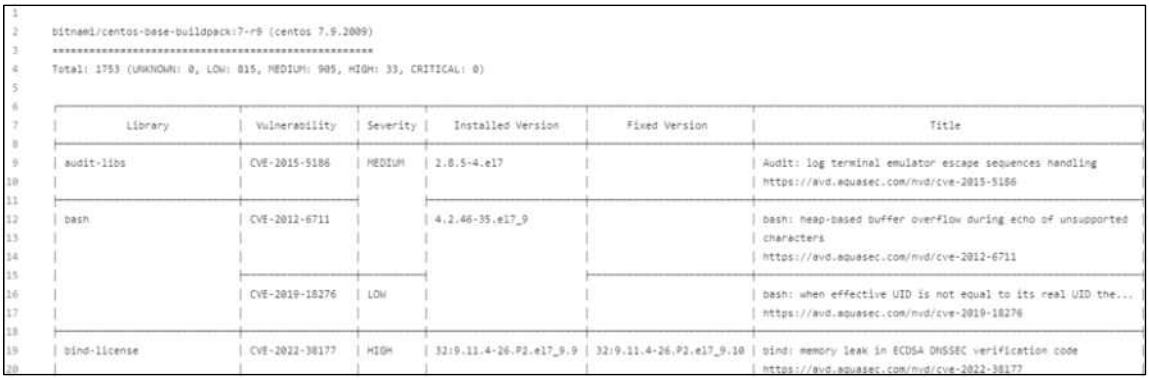
1) 사용자가 분석을 원하는 도커 이미지명을 입력하면 해당 도커 이미지는 DB에 저장된다.
2) DB에 새로운 이미지 요청이 들어올 경우, 분석 서버의 Client에서 내용을 확인한다.
3) 정적분석을 진행하고 실행 결과를 DB에 반환한다. 이를 웹 서버를 통해 사용자에게 전달한다.
4) 정적분석과 마찬가지로 동적분석 진행 후, 결과를 웹 서버를 통해 사용자에게 전달한다.
5) 정적/동적분석 결과를 바탕으로 결과 보고서를 생성하여 사용자에게 전달한다.
6) 이후 같은 이미지에 대한 분석 요청이 들어올 경우, DB에 있는 결과를 반환한다.



[그림 1 서비스 구성도]

3.4 정적분석

정적분석 단계에서는 Trivy를 통해 CVE 취약점 분석을 진행한다. Trivy를 사용하여 도커 이미지 취약점 분석을 진행할 시, 아래와 같이 이미지에서 발견된 CVE 정보를 얻을 수 있다.

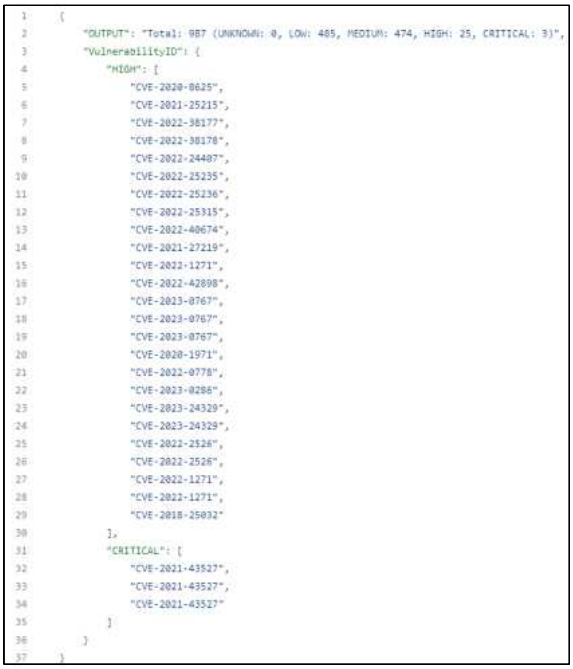


```
bitnami/centos-base-buildpack:7-r9 (centos 7.9.2009)
====================================================
Total: 1753 (UNKNOWN: 0, LOW: 815, MEDIUM: 905, HIGH: 33, CRITICAL: 0)

┌──────────────┬────────────────┬──────────┬──────────────────┬──────────────────┬──────────────────────────────────────────────────────────┐
│   Library    │ Vulnerability  │ Severity │ Installed Version│  Fixed Version   │                          Title                           │
├──────────────┼────────────────┼──────────┼──────────────────┼──────────────────┼──────────────────────────────────────────────────────────┤
│ audit-libs   │ CVE-2015-5186  │ MEDIUM   │ 2.8.5-4.el7      │                  │ Audit: log terminal emulator escape sequences handling   │
│              │                │          │                  │                  │ https://avd.aquasec.com/nvd/cve-2015-5186                │
├──────────────┼────────────────┼──────────┼──────────────────┼──────────────────┼──────────────────────────────────────────────────────────┤
│ bash         │ CVE-2012-6711  │          │ 4.2.46-35.el7_9  │                  │ bash: heap-based buffer overflow during echo of unsupported │
│              │                │          │                  │                  │ characters                                               │
│              │                │          │                  │                  │ https://avd.aquasec.com/nvd/cve-2012-6711                │
│              ├────────────────┼──────────┤                  ├──────────────────┼──────────────────────────────────────────────────────────┤
│              │ CVE-2019-18276 │ LOW      │                  │                  │ bash: when effective UID is not equal to its real UID the... │
│              │                │          │                  │                  │ https://avd.aquasec.com/nvd/cve-2019-18276               │
├──────────────┼────────────────┼──────────┼──────────────────┼──────────────────┼──────────────────────────────────────────────────────────┤
│ bind-license │ CVE-2022-38177 │ HIGH     │ 32:9.11.4-26.P2.el7_9.9 │ 32:9.11.4-26.P2.el7_9.10 │ bind: memory leak in ECDSA DNSSEC verification code │
│              │                │          │                  │                  │ https://avd.aquasec.com/nvd/cve-2022-38177               │
```

[그림 2 Trivy를 통한 취약점 분석 결과]

도커 이미지에서 발견된 CVE 총 개수와 CRITICAL, HIGH 레벨에 해당하는 CVE를 파싱하여 JSON 파일로 저장한 후, DB에 기록한다.




```
{
    "OUTPUT": "Total: 987 (UNKNOWN: 0, LOW: 485, MEDIUM: 474, HIGH: 25, CRITICAL: 3)",
    "VulnerabilityID": {
        "HIGH": [
            "CVE-2020-8625",
            "CVE-2021-25215",
            "CVE-2022-38177",
            "CVE-2022-38178",
            "CVE-2022-24407",
            "CVE-2022-25235",
            "CVE-2022-25236",
            "CVE-2022-25315",
            "CVE-2022-40674",
            "CVE-2021-27219",
            "CVE-2022-1271",
            "CVE-2022-42898",
            "CVE-2023-0767",
            "CVE-2023-0767",
            "CVE-2023-0767",
            "CVE-2020-1971",
            "CVE-2022-0778",
            "CVE-2023-0286",
            "CVE-2023-24329",
            "CVE-2023-24329",
            "CVE-2022-2526",
            "CVE-2022-2526",
            "CVE-2022-1271",
            "CVE-2022-1271",
            "CVE-2018-25032"
        ],
        "CRITICAL": [
            "CVE-2021-43527",
            "CVE-2021-43527",
            "CVE-2021-43527"
        ]
    }
}
```


[그림 3 정적분석 JSON 결과]

3.5 샌드박스

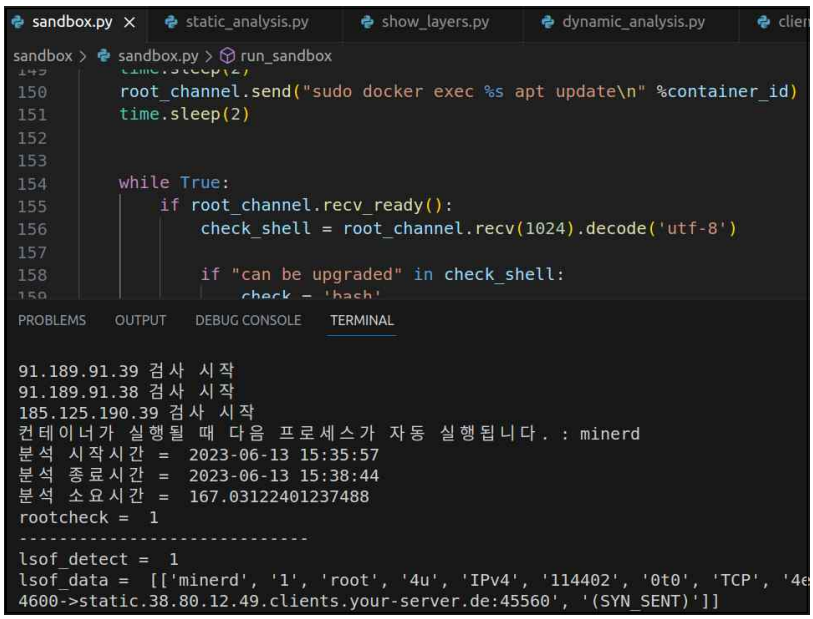
도커 이미지 동적분석 시, 악성 도커 이미지가 분석 PC를 감염시킬 우려가 있다. 이를 예방하기 위해 호스트 PC와 분리된 환경에서 안전하게 악성 도커 컨테이너를 실행시키고자 하였다. VirtualBox와 MongoDB에서 제공하는 Python API를 통해 도커 이미지를 가상 환경에서 실행하고, 결과를 DB에 저장하는 방식으로 구현하였다.

프로그램의 흐름은 아래와 같다.

1) Virtualbox API를 활용하여 Ubuntu 가상 머신을 실행시키고, SSH포트를 이용하여 원격 접속한다.

2) 가상 머신에서 분석하고자 하는 도커 컨테이너를 실행시킨다.

3) 동적분석을 수행한다.
4) 분석 결과를 반환하여 MongoDB API를 이용해 데이터베이스에 저장한다.



[그림 4 샌드박스를 통한 동적분석 자동화 코드 실행 결과]

3.6 동적분석

동적분석은 분석 서버의 샌드박스 내 컨테이너에서 동작한다. 동적분석으로는 Root 권한 실행 검사, 네트워크 검사, 열린 프로세스 검사, 자동 실행 파일 검사를 진행한다. 이러한 검사 프로세스가 자동화되어 진행되도록 Python 코드를 작성하였다.

3.6.1 Root 권한 실행 검사

컨테이너가 Root 계정으로 실행되면 해당 컨테이너에는 시스템 전체에 대한 권한이 부여된다. 이는 최소 권한 원칙을 위반하며, 컨테이너 내의 악의적인 코드나 공격자가 컨테이너에 액세스할 경우 시스템 전체에 대한 침투와 악용 가능성을 높일 수 있다.

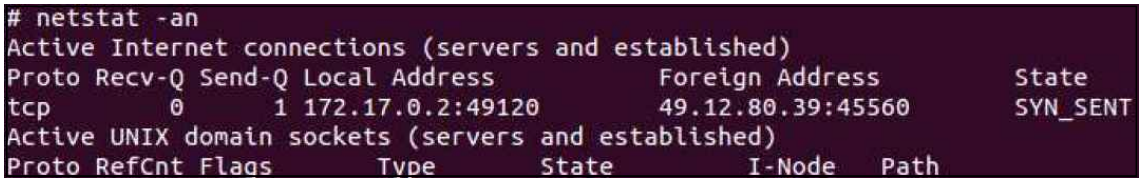
whoami 명령어를 통해 현재 컨테이너가 root 계정으로 실행되고 있는지 확인한다.

```
# whoami
root
```

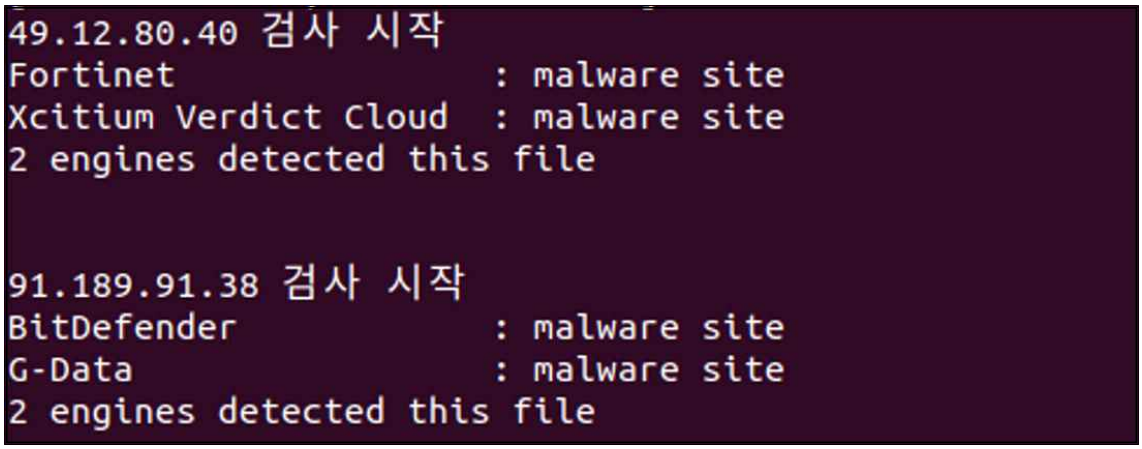
[그림 5 whoami 명령어 실행 결과]

3.6.2 네트워크 검사

외부 아이피와의 통신 탐지를 위해 netstat -an 명령어를 수행해 네트워크 연결 상태 및 현재 통신하는 아이피를 확인한다. 탐지된 아이피는 Virustotal을 통해 검사를 진행한다.



[그림 6 netstat -an 명령어 실행 결과]

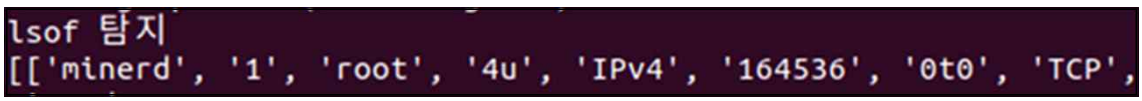


[그림 7 네트워크 검사 코드 실행 결과]

3.6.3 열린 프로세스 검사

비인가된 포트 및 프로세스는 시스템에 대한 악의적인 접근을 허용하게 한다. lsof -i TCP 명령어를 수행해 열린 TCP 포트 및 관련 프로세스 통신을 확인한다.

[그림 8 lsof -i TCP 명령어 실행 결과]



[그림 9 열린 프로세스 검사 코드 실행 결과]

3.6.4 자동 실행 파일 검사

docker history는 이미지가 만들어지기까지의 과정 전체를 보여주는 명령어이다. 즉, 이미지 레이어 전체를 볼 수 있다. docker history 명령어를 수행해 이미지 레이어를 추출하고, 레이어 중 ENTRYPOINT 명령으로 파일을 실행시키는 과정이 존재하는지 확인한다. ENTRYPOINT는 이미지가 컨테이너로 생성될 때, 실행되는 프로세스를 강제하는 옵션이다. 악성 도커 이미지는 ENTRYPOINT를 사용해 컨테이너 내에 악성 파일을 실행시키기도 한다. 이러한 특징을 이용해 ENTRYPOINT로 자동 실행되는 파일이 있다면 추출하고, 해당 파일은 Virustotal을 통해 검사를 진행한다.

[그림 10 docker history 명령어 실행 결과]

[그림 11 자동 실행 파일 검사 코드 실행 결과]

3.7 웹 구성

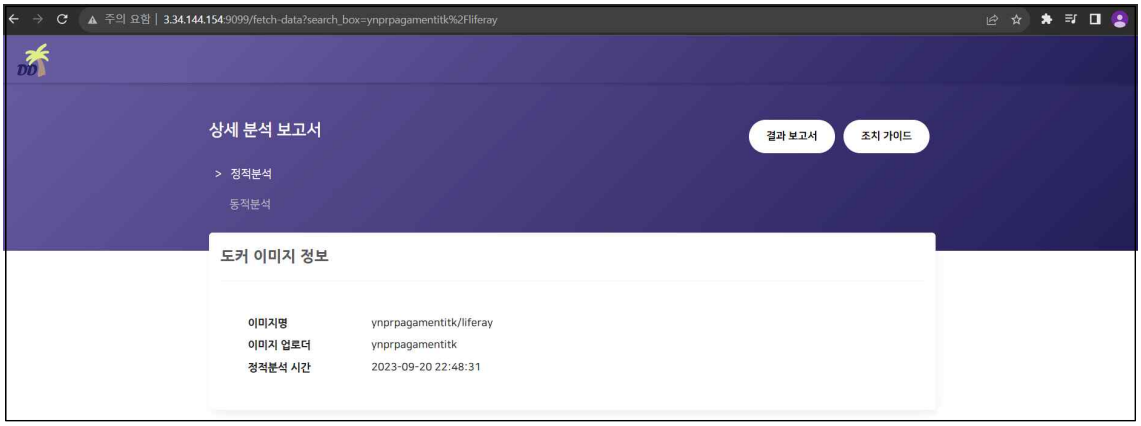
3.7.1 분석 보고서 검색 기능

사이트 메인 페이지에 위치한 검색창을 통해 도커 허브 내 도커 이미지의 분석 보고서를 검색할 수 있다. 도커 허브에 존재하지 않는 이미지를 검색할 경우 도커 허브에 존재하지 않는 이미지라는 알림창이 나타난다.

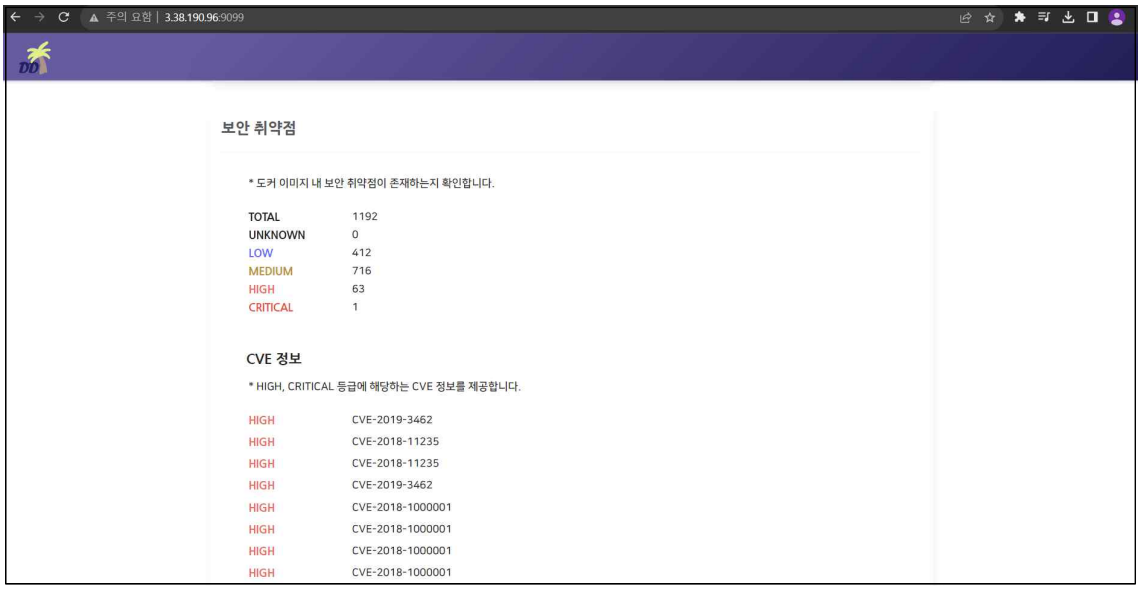
[그림 12 Docker Detect 메인]

3.7.2 상세 분석 보고서

각 도커 이미지에 대한 정적 및 동적 분석 보고서를 제공한다. 상세 분석 보고서는 각 분석 항목에 대한 간단한 설명 및 분석 결과, 위험도, 분석 결과에 따른 조치 방법으로 구성되어 있다. 결과 보고서 및 조치 가이드도 함께 다운로드 받을 수 있다.

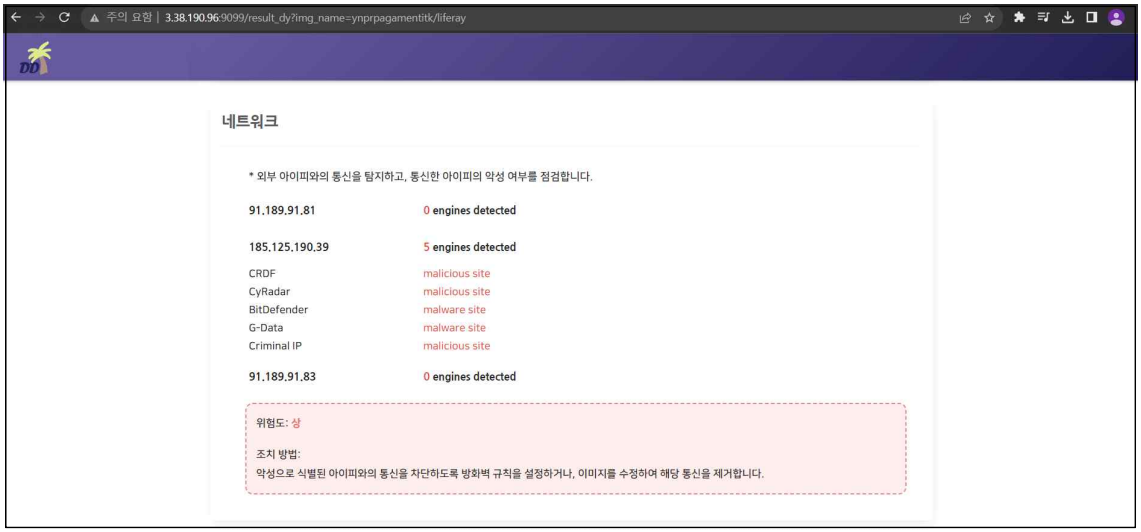


[그림 13 상세 분석 보고서 - 도커 이미지 정보]

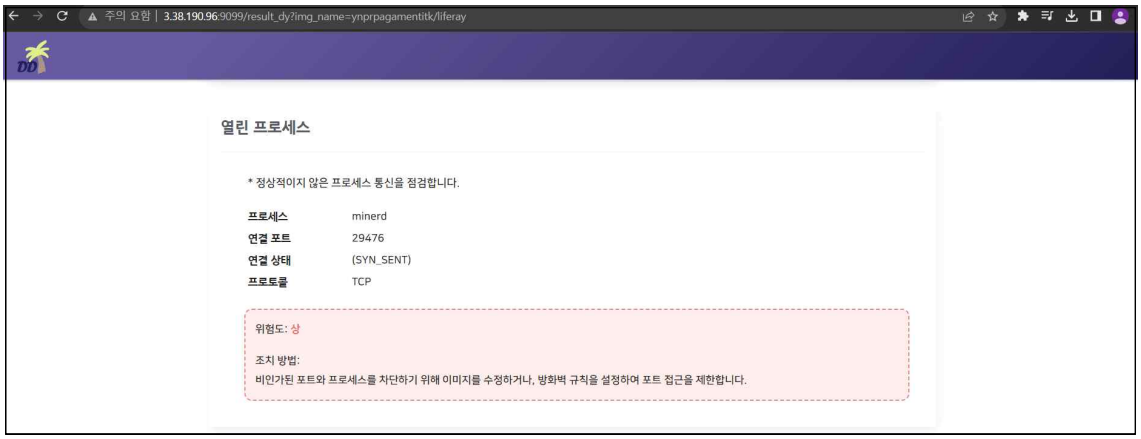


[그림 14 상세 분석 보고서 - 보안 취약점]

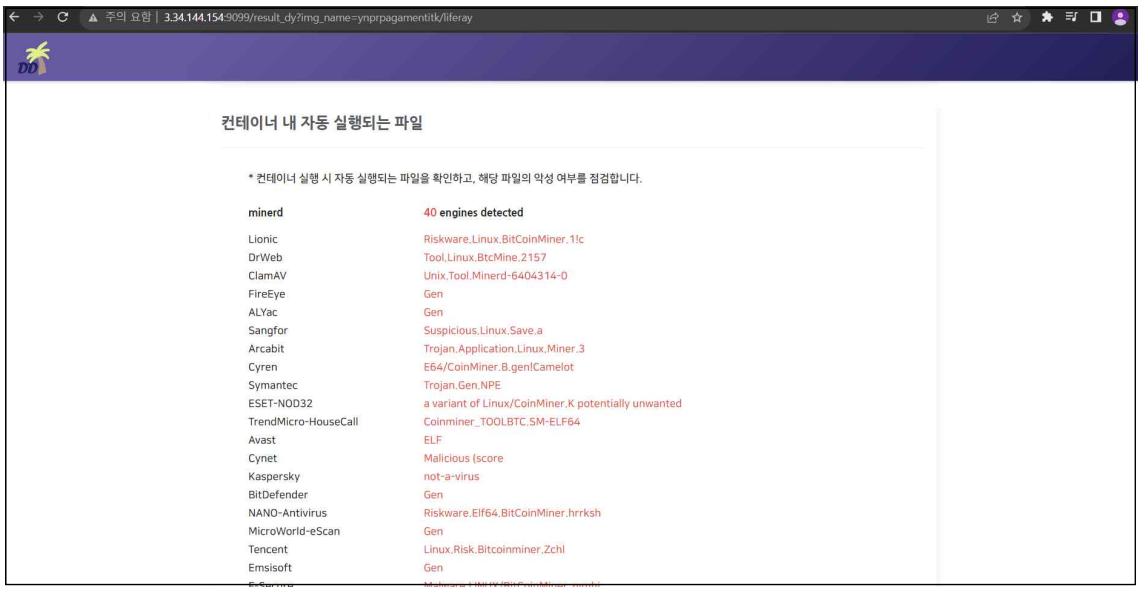
[그림 15 상세 분석 보고서 - Root 실행 여부]



[그림 16 상세 분석 보고서 - 네트워크]



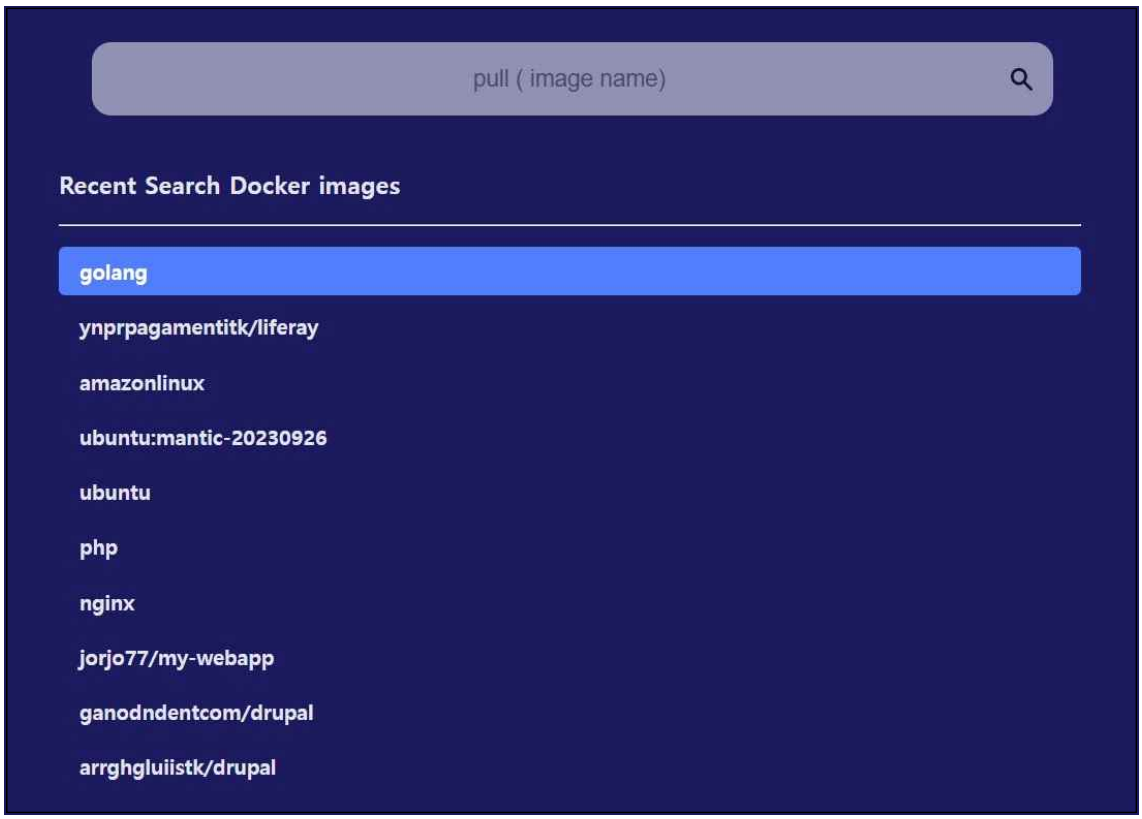
[그림 17 상세 분석 보고서 - 열린 프로세스]



[그림 18 상세 분석 보고서 - 컨테이너 내 자동 실행되는 파일]

3.7.3 최근 검색한 도커 이미지 리스트

메인 페이지에서 사용자들이 최근에 검색한 도커 이미지 리스트를 확인할 수 있다. 이미지명 클릭 시, 해당 이미지의 분석 보고서 페이지로 이동한다.



[그림 19 최근 검색한 도커 이미지 리스트]

3.8 컨설팅

컨설팅 전반적인 개요는 환경분석, 위험분석, 보안 대책 순서대로 진행된다. 각 단계에 대한 세부적인 내용은 아래와 같다.

3.8.1 환경분석

환경분석 단계에서는 요구 사항을 정의하고, 현황 분석을 진행하여 진단 범위를 확정한다. 이를 통해 컨설팅 목표가 수립된다.

2020년, 안랩에서 발표한 기사에 따르면 도커 허브라는 레포지토리에 올라가 있는 400만 개의 컨테이너 이미지를 보안 분석한 결과, 절반 이상 이미지에 최소 1개의 중대한 취약점이 포함되어 있다고 한다. 해당 프로젝트에서는 도커 허브에 있는 도커 이미지들이 오래된 소프트웨어 라이브러리를 사용하거나, 패치되지 않은 취약점을 내포하거나, 악성코드를 포함하여 배포하는 행위를 탐지하고자 했다.

국내외 기술 자료를 바탕으로 아래와 같이 점검 기준을 만들어, 진단 범위를 확정하였다.

구분	진단 코드	진단 항목
가. CVE 진단	DS 1-1	오래된 소프트웨어 구성
	DS 1-2	안전하지 않은 코딩
	DS 1-3	네트워크 보안
	DS 1-4	잘못 구성된 설정
	DS 1-5	약한 인증 및 액세스 제어
나. 권한 설정	DD 1-1-1	Root 계정: 탐지
	DD 1-1-2	Root 계정: 미탐지
다. 접근 통제	DD 2-1-1	외부 아이피 통신: 악성 아이피로 분류
	DD 2-1-2	외부 아이피 통신: 악성 아이피로 미분류
	DD 2-1-3	외부 아이피 통신: 미탐지
	DD 2-2-1	TCP 통신 프로세스: 탐지
	DD 2-2-2	TCP 통신 프로세스: 미탐지
라. 실행 파일 탐지	DD 3-1-1	자동 실행 파일: 악성코드로 분류
	DD 3-1-2	자동 실행 파일: 악성코드로 미분류
	DD 3-1-3	자동 실행 파일: 미탐지

[표 2 도커 이미지 취약점 분석·평가 항목]

3.8.2 위험분석

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

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

[표 3 위험도 구분]

산정된 위험도는 환경분석 단계에서 도출해낸 진단 항목에 맞추어 다음과 같이 적용하였다.

구분	진단 코드	진단 항목	위험도
가. CVE 진단	DS 1-1	오래된 소프트웨어 구성	*N/A*
	DS 1-2	안전하지 않은 코딩	*N/A*
	DS 1-3	네트워크 보안	*N/A*
	DS 1-4	잘못 구성된 설정	*N/A*
	DS 1-5	약한 인증 및 액세스 제어	*N/A*
나. 권한 설정	DD 1-1-1	Root 계정: 탐지	중
	DD 1-1-2	Root 계정: 미탐지	*N/A*
다. 접근 통제	DD 2-1-1	외부 아이피 통신: 악성 아이피로 분류	상
	DD 2-1-2	외부 아이피 통신: 악성 아이피로 미분류	중
	DD 2-1-3	외부 아이피 통신: 미탐지	하
	DD 2-2-1	TCP 통신 프로세스: 탐지	상
	DD 2-2-2	TCP 통신 프로세스: 미탐지	하
라. 실행 파일 탐지	DD 3-1-1	자동 실행 파일: 악성코드로 분류	상
	DD 3-1-2	자동 실행 파일: 악성코드로 미분류	중
	DD 3-1-3	자동 실행 파일: 미탐지	하

[표 4 도커 이미지 취약점 분석·평가 항목 위험도 적용]

3.8.3 보안 대책

각 항목에 대한 보안 대책은 아래와 같이 취약점 개요, 판단 기준 및 진단 방법 항목에 대해 작성하여 조치 가이드를 제작하였다.

DS 1-1	가. CVE 진단 > 오래된 소프트웨어 구성		위험도	N/A
취약점 개요				
점검 내용	■ 사용 중인 소프트웨어의 버전, 공급 업체의 지원 종료 여부, 보안 패치 및 업데이트 적용 여부 확인			
점검 목적	■ 최신 보안 패치 및 업데이트를 설치하여 취약점을 해결하고, 지원 종료된 소프트웨어의 사용을 방지하기 위함			
보안 위협	■ 이전에 발견된 취약점이 존재하거나, 새로운 보안 취약점에 대한 패치가 적용되지 않아 악용될 수 있음			
판단 기준 및 진단 방법				
판단 기준	**양호**	사용 중인 소프트웨어가 최신 버전이고 지원 및 보안 패치가 적용되는 경우		
	취약	오래된 버전 사용 중이거나 보안 패치가 적용되지 않은 경우		
진단 방법	■ 소프트웨어 버전 확인: 설치된 소프트웨어의 버전을 확인하고, 공급 업체의 웹 사이트나 업데이트 정보를 확인하여 최신 버전인지 확인함 ■ 공급 업체 공지 사항 및 업데이트 정보 확인함 ■ 취약성 데이터베이스 및 보안 패치 확인을 통해서 진단			
조치 방법				
조치 방법	■ 최신 보안 패치와 업데이트를 적용 ■ 사용 중인 소프트웨어가 지원 종료됐다면 대안 소프트웨어나 업그레이드를 고려함			

[표 5 조치 가이드]

3.8.4 결과 보고서

위 진단 항목을 토대로 도커 이미지에 대한 정적, 동적 분석이 진행된 후, 해당 결과를 보고서 형태로 확인할 수 있게 자동화하였다. 사용자는 결과 보고서를 통해 수행된 항목 목록별 탐지된 취약점 개수와 탐지 결과를 확인할 수 있다.

스캔 정보

Image name: ynprpagamentitk/liferay

Start time: 2023-06-12 19:10:29
Finish time: 2023-06-12 19:12:41
Elapsed: 130sec

수행된 항목 목록

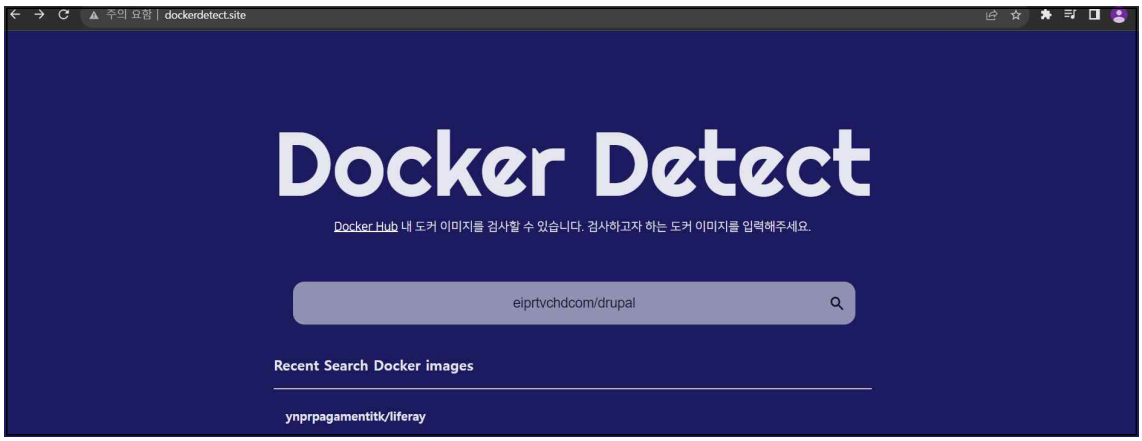
구분	진단 항목	개수/결과
CVE 진단	CVE Critical	1
	CVE High	63
	CVE Medium	717
	CVE Low	412
	CVE Unknown	0
권한 설정	Root 계정	탐지
접근 통제	외부 아이피 통신	탐지
	TCP 통신 프로세스	탐지
실행 파일 탐지	자동 실행 파일	탐지

[그림 20 도커 이미지 취약성 결과 보고서]

4. 분 석

4.1 서비스 활용 안내 및 사례

Docker Detect 서비스 활용 결과는 아래와 같다. 도커 허브에서 도커 이미지를 Pull하여 내려받기 전에 먼저 Docker Detect 사이트에 해당 이미지를 검색한다.



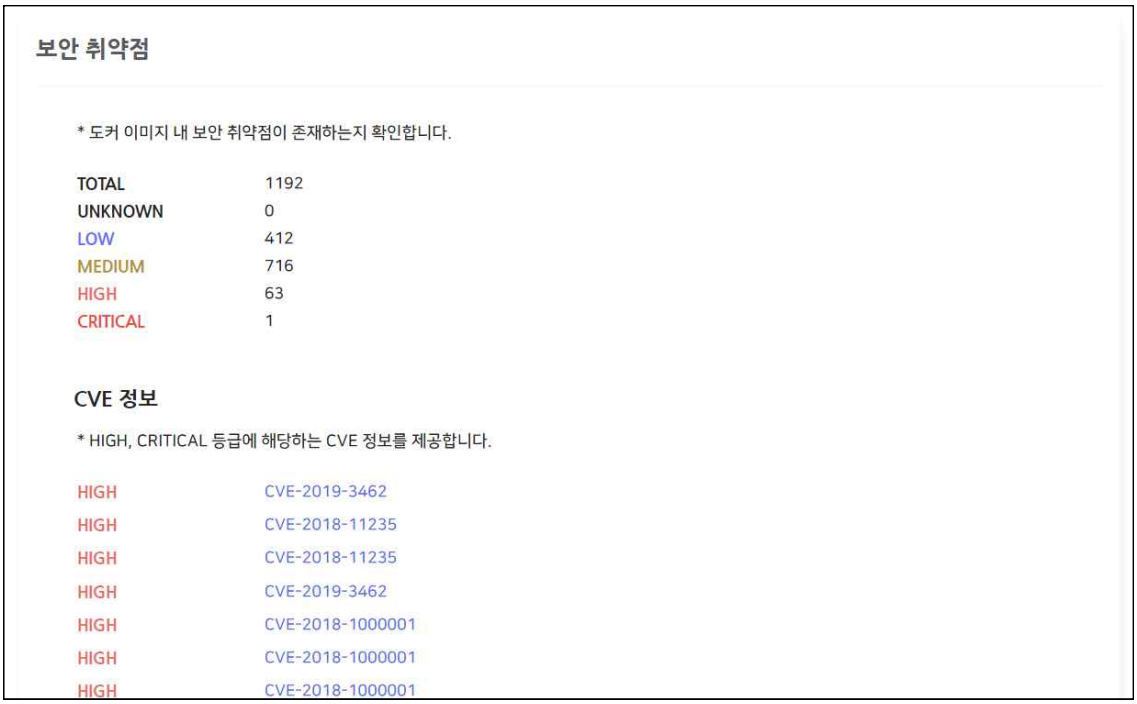
[그림 21 Docker Detect 메인]

이미지를 검색하면 상세 분석 보고서 페이지로 이동된다. 정적분석 보고서에서는 도커

이미지 정보, 보안 취약점 검사 결과를 확인할 수 있다. 도커 이미지 정보에는 이미지명, 이미지 업로더, 정적분석 시간이 표시되어 있다.

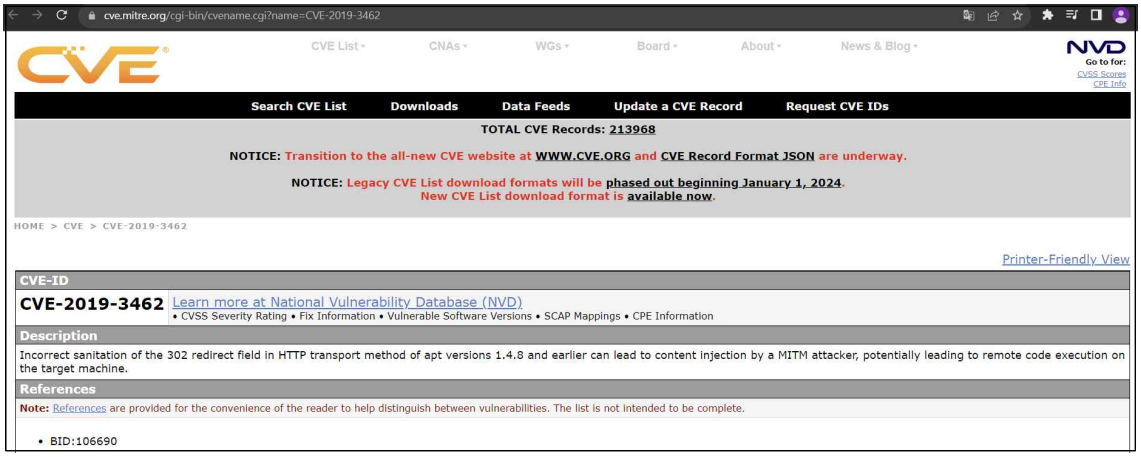
[그림 22 도커 이미지 정보]

보안 취약점 검사 결과에서는 도커 이미지 내 전체 취약점 개수와 취약점 등급별 개수를 확인할 수 있으며, HIGH, CRITICAL 등급에 해당하는 CVE를 확인할 수 있다.



[그림 23 상세 분석 보고서 - 보안 취약점]

각 CVE를 클릭하면 해당 CVE에 대한 상세 정보를 볼 수 있다.



[그림 24 CVE 상세 정보(cve.mitre.org)]

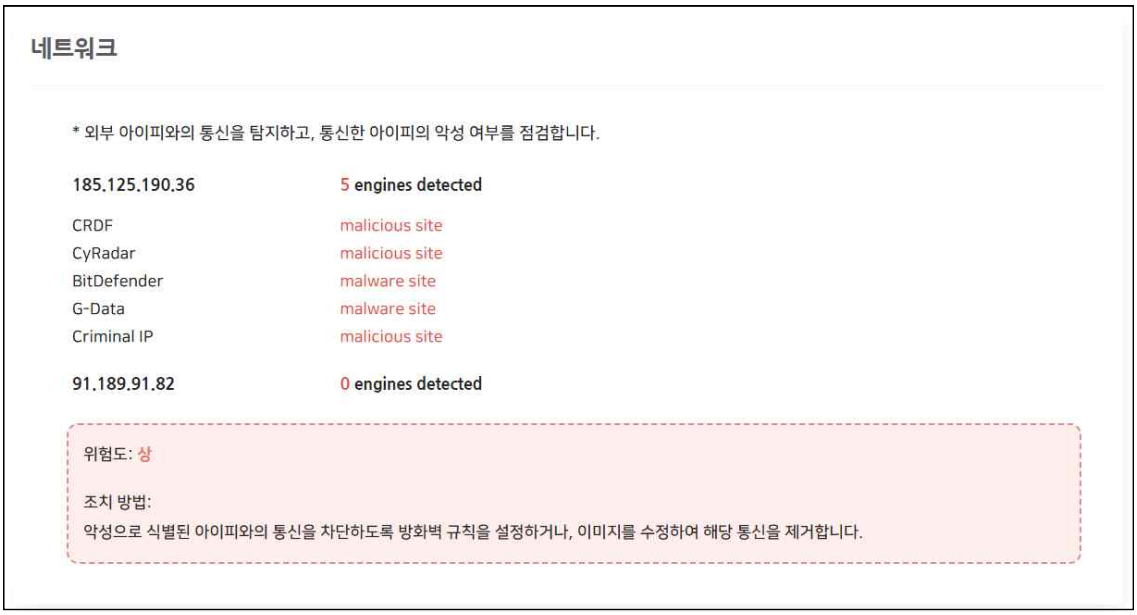
동적분석 보고서에서는 도커 이미지 정보, Root 실행 여부 검사, 네트워크 검사 결과, 열린 프로세스 검사 결과, 컨테이너 내 자동 실행되는 파일 검사 결과를 확인할 수 있다. 도커 이미지 정보에는 정적분석 내 도커 이미지 정보와 마찬가지로 이미지명과 이미지 업로더, 동적분석 시간이 표시되어 있다

[그림 25 상세 분석 보고서 - 도커 이미지 정보]

Root 실행 여부 검사 결과에서는 컨테이너 실행 시 기본 Shell이 Root 권한으로 실행되는지 확인할 수 있다. 동적분석 보고서는 각 항목 결과에 따른 위험도와 그에 대한 조치 방법도 함께 제공한다. 참고하여 해당 도커 이미지 내 어떤 위험이 존재하고, 어떻게 대처해야 하는지를 파악할 수 있다.

[그림 26 상세 분석 보고서 - Root 실행 여부]

네트워크 검사 결과에서는 컨테이너 내에서 외부 아이피와 통신이 이뤄지는지, 해당 아이피가 악성인지를 확인할 수 있다.

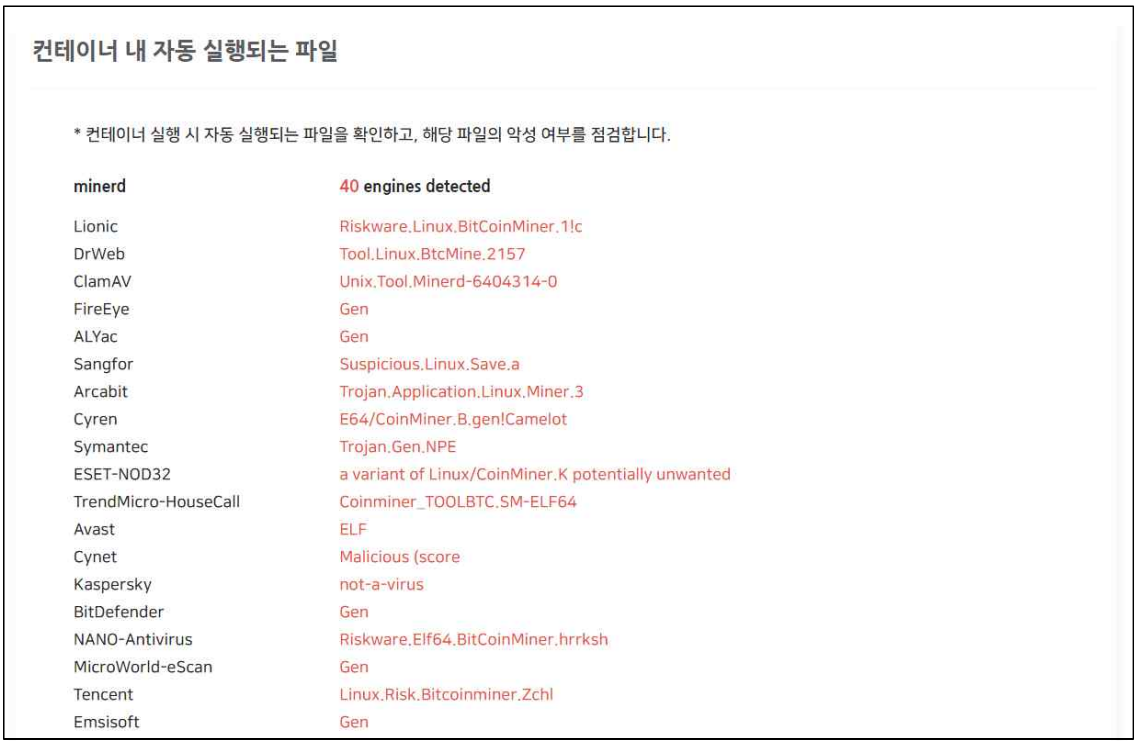


[그림 27 상세 분석 보고서 - 네트워크]

열린 프로세스 검사 결과에서는 컨테이너 내 프로세스 통신을 확인할 수 있으며, 연결된 프로세스의 연결 포트, 연결 상태, 프로토콜 정보를 알 수 있다.

[그림 28 상세 분석 보고서 - 열린 프로세스]

컨테이너 내 자동 실행되는 파일 검사 결과에서는 컨테이너 내에서 자동으로 실행되는 파일이 존재하는지, 해당 파일이 악성인지를 확인할 수 있다.



[그림 29 상세 분석 보고서 – 컨테이너 내 자동 실행되는 파일]

결과보고서에는 도커 이미지에 대한 분석 결과가 나와 있다. CVE 진단 결과 및 각 동적 분석 항목별 탐지/미탐지 결과를 확인할 수 있다.

[그림 30 결과보고서]

조치 가이드에서는 도커 이미지 취약점 분석·평가 항목 체크 리스트, 위험도 산정 기준, 정적분석과 동적분석 항목에 따른 상세한 진단 내용과 조치 방법을 확인할 수 있다.

중부대학교 정보보호프로젝트실습1(캡스톤디자인)

Team.야자나무

도커 이미지

기술적 취약점 정적·동적 진단 상세 조치 가이드

2023.06

I. 전체목록

1. 체크리스트 항목

도커 이미지 취약점 진단에 사용될 체크리스트는 국내외 기술 자료를 바탕으로 작성하였다.

가이드 내 영역은 크게 정적 분석, 동적 분석으로 구성하였으며, 정적 분석은 CVE 진단(5개 항목), 동적 분석은 권한 설정(1개 항목), 접근 통제(2개 항목), 리소스 관리(1개 항목)로 총 9개 항목으로 제작하였다.

구분	진단 코드	진단 항목	위험도
가. CVE 진단	DS 1-1	오래된 소프트웨어 구성	N/A
	DS 1-2	안전하지 않은 코딩	N/A
	DS 1-3	네트워크 보안	N/A
	DS 1-4	잘못 구성된 설정	N/A
	DS 1-5	약한 인증 및 액세스 제어	N/A
나. 권한 설정	DD 1-1-1	Root 계정: 탐지	중
	DD 1-1-2	Root 계정: 미탐지	N/A
다. 접근 통제	DD 2-1-1	외부 아이피 통신: 악성 아이피로 분류	상
	DD 2-1-2	외부 아이피 통신: 악성 아이피로 미분류	중
	DD 2-1-3	외부 아이피 통신: 미탐지	하
	DD 2-2-1	TCP 통신 프로세스: 탐지	상
	DD 2-2-2	TCP 통신 프로세스: 미탐지	하
라. 실행 파일 탐지	DD 3-1-1	자동 실행 파일: 악성코드로 분류	상
	DD 3-1-2	자동 실행 파일: 악성코드로 미분류	중
	DD 3-1-3	자동 실행 파일: 미탐지	하

[표 2] 도커 이미지 취약점 분석·평가 항목

2. 위험도

잠재적 위협이 존재하는 취약점의 위험도를 상, 중, 하 3단계로 구분하였다.

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

[표 3] 위험도 구분

II. 세부 항목

1. 정적 분석

1.1. CVE 진단

DS 1-1	가. CVE 진단 > 오래된 소프트웨어 구성	위험도	N/A
취약점 개요			
점검 내용	■ 사용 중인 소프트웨어의 버전, 공급 업체의 지원 종료 여부, 보안 패치 및 업데이트 적용 여부 확인		
점검 목적	■ 최신 보안 패치 및 업데이트를 설치하여 취약점을 해결하고, 지원 종료된 소프트웨어의 사용을 방지하기 위함		
보안 위협	■ 이전에 발견된 취약점이 존재하거나, 새로운 보안 취약점에 대한 패치가 적용되지 않아 악용될 수 있음		
판단 기준 및 진단 방법			
판단 기준	양호	사용 중인 소프트웨어가 최신 버전이고 지원 및 보안 패치가 적용되는 경우	
	취약	오래된 버전 사용 중이거나 보안 패치가 적용되지 않은 경우	
진단 방법	■ 소프트웨어 버전 확인: 설치된 소프트웨어의 버전을 확인하고, 공급 업체의 웹 사이트나 업데이트 정보를 확인하여 최신 버전인지 확인함 ■ 공급 업체 공지 사항 및 업데이트 정보 확인함 ■ 취약성 데이터베이스 및 보안 패치 확인을 통해서 진단		
조치 방법			
조치 방법	■ 최신 보안 패치와 업데이트를 적용 ■ 사용 중인 소프트웨어가 지원 종료됐다면 대안 소프트웨어나 업그레이드를 고려함		

[표 4] 세부 항목 설정

DS 1-2	가. CVE 진단 > 안전하지 않은 코딩	위험도	N/A
취약점 개요			
점검 내용	■ 애플리케이션의 소스 코드를 검토하여 보안 취약점이 있는지 확인		
점검 목적	■ 악의적인 사용자로부터의 공격에 취약할 수 있으므로, 이를 보완하기 위함		
보안 위협	■ 버퍼 오버플로우, 인증 및 인가 오류, RCE 등 다양한 취약점을 유발할 수 있음		
판단 기준 및 진단 방법			
판단 기준	양호	안전한 코딩 관행을 따르며, 취약점을 찾아 수정한 경우	
	취약	취약점이 발견되거나 안전하지 않은 코딩 관행이 적용되지 않은 경우	
진단 방법	■ 정적 코드 분석, 동적 코드 분석, 취약성 스캐닝 도구, 수동 코드 검토 등을 통하여 애플리케이션의 소스 코드를 분석함		
조치 방법			
조치 방법	■ 안전한 코딩 관행을 따르고, 취약점을 수정하고 보완하기 위해 코딩 및 테스트 프로세스를 개선함		

[표 5] 세부 항목 설정

DS 1-3	가. CVE 진단 > 네트워크 보안	위험도	N/A
취약점 개요			
점검 내용	■ 네트워크 구성, 방화벽 설정, 암호화 프로토콜 등을 검토하여 네트워크 보안 확인		
점검 목적	■ 올바른 네트워크 보안 구성을 통한 데이터 유출, 악성 트래픽, 액세스 제어 위반 등을 방지하기 위함		
보안 위협	■ 부적절한 방화벽 설정, 암호화되지 않은 트래픽, 네트워크 구성 오류 등으로 인한 중요 데이터 유출 및 악용 가능성을 초래함		
판단 기준 및 진단 방법			
판단 기준	양호	적절한 네트워크 구성 및 방화벽 설정, 암호화 프로토콜을 사용하는 경우	
	취약	부적절한 방화벽 설정, 암호화되지 않은 통신, 취약한 포트 등 발견된 경우	
진단 방법	■ 네트워크 구성 및 방화벽 설정 검토, 포트 스캐닝, 트래픽 분석 등을 통해 네트워크 보안을 평가함		
조치 방법			
조치 방법	■ 방화벽 설정 개선, 암호화 프로토콜의 도입, 취약한 포트의 제한 등을 통해 네트워크 보안을 강화함		

[표 6] 세부 항목 설정

DS 1-4	가. CVE 진단 > 잘못 구성된 설정	위험도	N/A
취약점 개요			
점검 내용	■ 시스템, 애플리케이션, 데이터베이스 등의 구성 설정을 검토하여 취약한 설정 확인		
점검 목적	■ 올바르게 구성된 설정을 통해 시스템의 안전성과 보안을 보장하기 위함		
보안 위협	■ 기본 비밀번호 사용, 약한 액세스 권한 설정, 기본 보안 설정의 미사용 등으로 인해 시스템에 대한 무단 액세스 및 데이터 유출 등을 초래함		
판단 기준 및 진단 방법			
판단 기준	양호	기본 비밀번호 변경, 적절한 액세스 권한 설정, 보안 설정 활성화 등이 확인된 경우	
	취약	기본 비밀번호 사용, 약한 액세스 권한 설정, 기본 보안 설정 미사용 등이 발견된 경우	
진단 방법	■ 설정 파일 검토, 권한 및 접근 제어 목록(Access Control List) 분석 등을 사용하여 취약한 설정을 확인함		
조치 방법			
조치 방법	■ 기본 비밀번호 변경, 적절한 액세스 권한 설정, 보안 설정 활성화 등을 통해 취약한 설정을 보완함		

[표 7] 세부 항목 설정

DS 1-5	가. CVE 진단 > 약한 인증 및 액세스 제어	위험도	N/A
취약점 개요			
점검 내용	■ 사용자 인증 및 액세스 제어 메커니즘을 검토하여 보안 결함 확인		
점검 목적	■ 인가되지 않은 액세스로부터 시스템과 데이터를 보호하여 시스템의 기밀성과 무결성을 보호하기 위함		
보안 위협	■ 약한 암호 정책, 기본 자격 증명 사용, 액세스 권한 부여의 부적절한 구성 등은 인증 및 액세스 제어에 취약점을 초래함		
판단 기준 및 진단 방법			
판단 기준	양호	강력한 암호 정책, 다중 인증 요구, 적절한 액세스 권한 제어 등이 확인된 경우	
	취약	약한 암호 정책, 잘못된 권한 부여, 취약한 인증 메커니즘 등이 발견된 경우	
진단 방법	■ 인증 메커니즘 검토, 암호 정책 분석, 사용자 계정 권한 분석 등을 통해 약한 인증 및 액세스 제어를 평가함		
조치 방법			
조치 방법	■ 강력한 암호 정책 설정, 다중 인증 요구, 적절한 액세스 권한 제어, 안전한 인증 메커니즘 구현 등을 통해 인증 및 액세스 제어를 강화함		

[표 8] 세부 항목 설정

2. 동적 분석

2.1. Root 권한 확인

DD 1-1-1	나. 권한 설정 > Root 계정: 탐지	위험도	중
취약점 개요			
점검 내용	■ 컨테이너 실행 시 별도의 관리자 계정 실행 및 로그인 가능 여부 점검		
점검 목적	■ 알려진 계정을 통한 비인가자의 무단 접근 시도를 예방하기 위함		
보안 위협	■ root 계정은 누구나 알 수 있는 계정이기 때문에, 비인가자의 접속 시도 및 비밀번호 무작위 대입 공격에 노출될 수 있음		
판단 기준 및 진단 방법			
판단 기준	■ 컨테이너가 root 계정으로 실행 및 로그인되고 있는 경우		

진단 방법	■ 아래 명령어를 통해 현재 컨테이너가 root 계정으로 실행되고 있는지 확인
	# whoami
	`# whoaml`
	`root`		

조치 방법			
조치 방법	■ root 계정 사용이 필요한 경우를 내부적으로 정의하여 정책 수립 ■ 주기적으로 root 계정 사용 여부를 점검하여 의도하지 않은 사용 탐지 필요		

[표 9] 세부 항목 설정

DD 1-1-2	나. 권한 설정 > Root 계정: 미탐지	위험도	N/A
취약점 개요			
점검 내용	■ 컨테이너 실행 시 별도의 관리자 계정 실행 및 로그인 가능 여부 점검		
점검 목적	■ 알려진 계정을 통한 비인가자의 무단 접근 시도를 예방하기 위함		
보안 위협	■ root 계정은 누구나 알 수 있는 계정이기 때문에, 비인가자의 접속 시도 및 비밀번호 무작위 대입 공격에 노출될 수 있음		
판단 기준 및 진단 방법			
판단 기준	■ 컨테이너가 root 계정으로 실행되거나 로그인되지 않은 경우		
진단 방법	■ 아래 명령어를 통해 현재 컨테이너가 root가 아닌 일반 계정으로 실행되고 있는지 확인 # whoami		
조치 방법			
조치 방법			

[표 10] 세부 항목 설정

2.2. 네트워크 검사

DD 2-1-1	다. 접근 통제 > 외부 아이피 통신: 악성 아이피로 분류	위험도	상
취약점 개요			
점검 내용	■ 외부 아이피와의 통신 탐지 및 통신한 아이피의 악성 여부 점검		
점검 목적	■ 악성 아이피 통신으로 인한 악성코드 다운로드 등 문제를 방지하기 위함		
보안 위협	■ 시스템 해킹, 악성코드 전파, 개인정보 유출 등의 문제를 초래할 수 있음		
판단 기준 및 진단 방법			
판단 기준	■ 외부 아이피와 통신 및 해당 아이피가 악성으로 식별되는 경우		

진단 방법	■ 아래 명령어를 통해 네트워크 연결 상태 확인 및 현재 통신하는 아이피 확인
	# netstat -an
	`# netstat -an`
	`Active Internet connections (servers and established)`
	`Proto Recv-Q Send-Q Local Address Foreign Address State`
	`tcp 0 1 172.17.0.2:49120 49.12.80.39:45560 SYN_SENT`
	`Active UNIX domain sockets (servers and established)`
	`Proto RefCnt Flags Type State I-Node Path`
	■ 검출된 외부 아이피에 대해 VirusTotal를 이용하여 진단 후, 악성 아이피로 확인됨		

조치 방법			
조치 방법	■ 악성으로 식별된 아이피와의 통신을 차단하도록 방화벽 규칙을 설정하거나, 이미지를 수정하여 해당 통신을 제거함		

[표 11] 세부 항목 설정

DD 2-1-2	다. 접근 통제 > 외부 아이피 통신: 악성 아이피로 미분류	위험도	중
취약점 개요			
점검 내용	■ 외부 아이피와 통신하는지 탐지 및 통신한 아이피의 악성 여부 점검		
점검 목적	■ 악성 아이피 통신으로 인한 악성코드 다운로드 등 문제를 진단하기 위함		
보안 위협	■ 시스템 해킹, 악성코드 전파, 개인정보 유출 등의 문제를 초래할 수 있음		
판단 기준 및 진단 방법			
판단 기준	■ 외부 아이피와 통신하나, 해당 아이피가 악성으로 식별되지 않은 경우		

진단 방법	■ 아래 명령어를 통해 네트워크 연결 상태 확인 및 현재 통신하는 아이피 확인
	# netstat -an
	`# netstat -an`
	`Active Internet connections (servers and established)`
	`Proto Recv-Q Send-Q Local Address Foreign Address State`
	`tcp 0 1 172.17.0.2:49120 49.12.80.39:45560 SYN_SENT`
	`Active UNIX domain sockets (servers and established)`
	`Proto RefCnt Flags Type State I-Node Path`
	■ 검출된 외부 아이피에 대해 VirusTotal를 이용하여 진단 후, 악성 아이피로 확인되지 않음		

조치 방법			
조치 방법	■ 악성으로 식별되지 않은 아이피라도 통신을 차단하도록 방화벽 규칙을 설정하거나, 이미지를 수정하여 해당 통신을 제거함		

[표 12] 세부 항목 설정

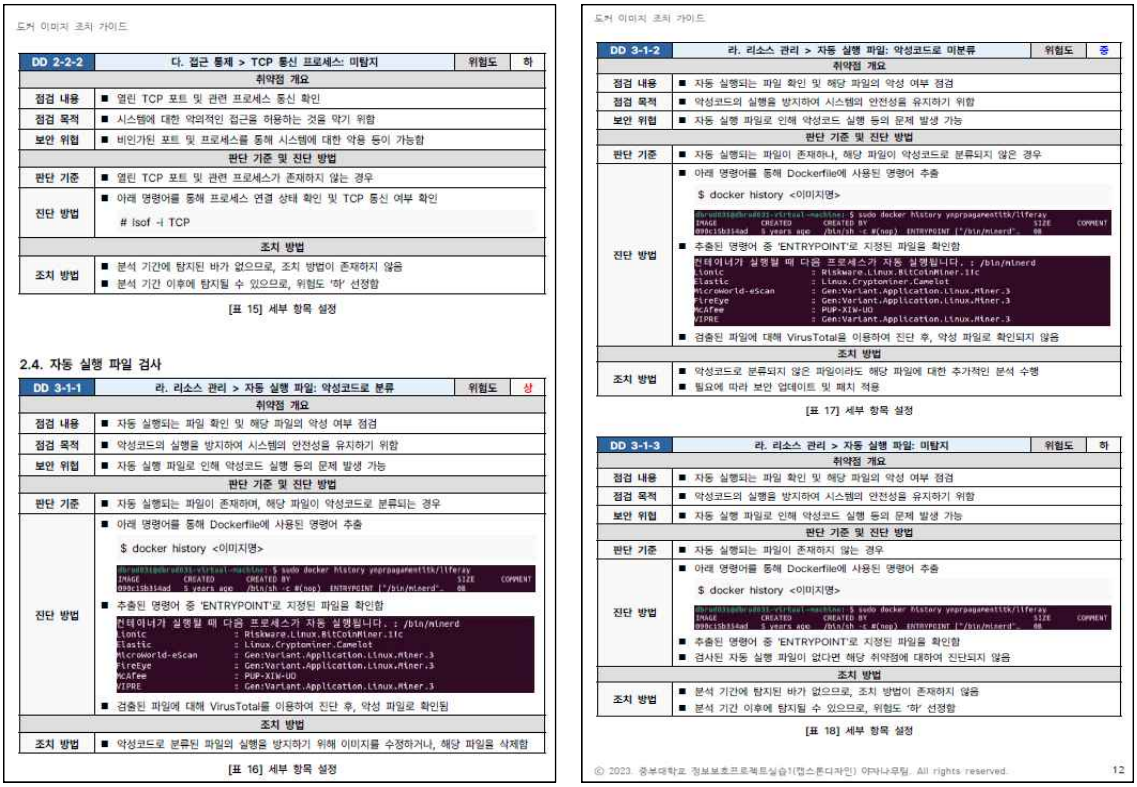
DD 2-1-3	다. 접근 통제 > 외부 아이피 통신: 미탐지	위험도	하
취약점 개요			
점검 내용	■ 외부 아이피와의 통신 탐지 및 통신한 아이피의 악성 여부 점검		
점검 목적	■ 악성 아이피 통신으로 인한 악성코드 다운로드 등 문제를 진단하기 위함		
보안 위협	■ 시스템 해킹, 악성코드 전파, 개인정보 유출 등의 문제를 초래할 수 있음		
비고			
판단 기준	■ 외부 아이피와 통신이 이뤄지지 않는 경우		
진단 방법	■ 아래 명령어를 통해 네트워크 연결 상태 확인 및 현재 통신하는 아이피 확인 # netstat -an ■ 검출된 외부 아이피가 없다면 해당 취약점에 대하여 진단되지 않음		
조치 방법			
조치 방법	■ 분석 기간에 탐지된 바가 없으므로, 조치 방법이 존재하지 않음 ■ 분석 기간 이후에 탐지될 수 있으므로, 위험도 '하' 선정함		

[표 13] 세부 항목 설정

2.3. 열린 프로세스 검사

DD 2-2-1	다. 접근 통제 > TCP 통신 프로세스: 탐지	위험도	상
취약점 개요			
점검 내용	■ 열린 TCP 포트 및 관련 프로세스 통신 확인		
점검 목적	■ 비인가된 포트 및 프로세스로 인해 시스템에 대한 악의적인 접근을 허용하는 것을 막기 위함		
보안 위협	■ 비인가된 포트 및 프로세스를 통해 시스템에 대한 악용 등이 가능함		
판단 기준 및 진단 방법			
판단 기준	■ 열린 TCP 포트 및 관련 프로세스가 존재하는 경우		
진단 방법	■ 아래 명령어를 통해 프로세스 연결 상태 확인 및 TCP 통신 여부 확인 # lsof -i TCP		
조치 방법			
조치 방법	■ 비인가된 포트와 프로세스를 차단하기 위해 이미지를 수정하거나, 방화벽 규칙을 설정하여 포트 접근을 제한함		

[표 14] 세부 항목 설정



도커 이미지 조치 가이드

DD 2-2-2	다. 접근 통제 > TCP 통신 프로세스: 미탐지	위험도	하
취약점 개요			
점검 내용	■ 열린 TCP 포트 및 관련 프로세스 통신 확인		
점검 목적	■ 시스템에 대한 악의적인 접근을 허용하는 것을 막기 위함		
보안 위협	■ 비인가된 포트 및 프로세스를 통해 시스템에 대한 악용 등이 가능함		
판단 기준 및 진단 방법			
판단 기준	■ 열린 TCP 포트 및 관련 프로세스가 존재하지 않는 경우		
진단 방법	■ 아래 명령어를 통해 프로세스 연결 상태 확인 및 TCP 통신 여부 확인 # lsof -i TCP		
조치 방법			
조치 방법	■ 분석 기간에 탐지된 바가 없으므로, 조치 방법이 존재하지 않음 ■ 분석 기간 이후에 탐지될 수 있으므로, 위협도 '하' 선정함		

[표 15] 세부 항목 설정

2.4. 자동 실행 파일 검사

DD 3-1-1	라. 리소스 관리 > 자동 실행 파일: 악성코드로 분류	위험도	상
취약점 개요			
점검 내용	■ 자동 실행되는 파일 확인 및 해당 파일의 악성 여부 점검		
점검 목적	■ 악성코드의 실행을 방지하여 시스템의 안전성을 유지하기 위함		
보안 위협	■ 자동 실행 파일로 인해 악성코드 실행 등의 문제 발생 가능		
판단 기준 및 진단 방법			
판단 기준	■ 자동 실행되는 파일이 존재하며, 해당 파일이 악성코드로 분류되는 경우		
진단 방법	■ 아래 명령어를 통해 Dockerfile에 사용된 명령어 추출 $ docker history <이미지명> ■ 추출된 명령어 중 'ENTRYPOINT'로 지정된 파일을 확인함 ■ 검출된 파일에 대해 VirusTotal를 이용하여 진단 후, 악성 파일로 확인됨		
조치 방법			
조치 방법	■ 악성코드로 분류된 파일의 실행을 방지하기 위해 이미지를 수정하거나, 해당 파일을 삭제함		

[표 16] 세부 항목 설정

DD 3-1-2	라. 리소스 관리 > 자동 실행 파일: 악성코드로 미분류	위험도	중
취약점 개요			
점검 내용	■ 자동 실행되는 파일 확인 및 해당 파일의 악성 여부 점검		
점검 목적	■ 악성코드의 실행을 방지하여 시스템의 안전성을 유지하기 위함		
보안 위협	■ 자동 실행 파일로 인해 악성코드 실행 등의 문제 발생 가능		
판단 기준 및 진단 방법			
판단 기준	■ 자동 실행되는 파일이 존재하나, 해당 파일이 악성코드로 분류되지 않은 경우		
진단 방법	■ 아래 명령어를 통해 Dockerfile에 사용된 명령어 추출 $ docker history <이미지명> ■ 추출된 명령어 중 'ENTRYPOINT'로 지정된 파일을 확인함 ■ 검출된 파일에 대해 VirusTotal를 이용하여 진단 후, 악성 파일로 확인되지 않음		
조치 방법			
조치 방법	■ 악성코드로 분류되지 않은 파일이라도 해당 파일에 대한 추가적인 분석 수행 ■ 필요에 따라 보안 업데이트 및 패치 적용		

[표 17] 세부 항목 설정

DD 3-1-3	라. 리소스 관리 > 자동 실행 파일: 미탐지	위험도	하
취약점 개요			
점검 내용	■ 자동 실행되는 파일 확인 및 해당 파일의 악성 여부 점검		
점검 목적	■ 악성코드의 실행을 방지하여 시스템의 안전성을 유지하기 위함		
보안 위협	■ 자동 실행 파일로 인해 악성코드 실행 등의 문제 발생 가능		
판단 기준 및 진단 방법			
판단 기준	■ 자동 실행되는 파일이 존재하지 않는 경우		
진단 방법	■ 아래 명령어를 통해 Dockerfile에 사용된 명령어 추출 $ docker history <이미지명> ■ 추출된 명령어 중 'ENTRYPOINT'로 지정된 파일을 확인함 ■ 검사된 자동 실행 파일이 없다면 해당 취약점에 대하여 진단되지 않음		
조치 방법			
조치 방법	■ 분석 기간에 탐지된 바가 없으므로, 조치 방법이 존재하지 않음 ■ 분석 기간 이후에 탐지될 수 있으므로, 위협도 '하' 선정함		

[표 18] 세부 항목 설정

[그림 31 조치 가이드]

4.2 추후 보완사항

도커허브를 통한 악성 도커 이미지 유포는 계속해서 증가하고 있다. 이에 따라 향후에 악성 도커 이미지를 지속적으로 분석하여 새로운 방식의 악성 도커 이미지도 탐지할 수 있도록 동적분석 기능을 업데이트할 필요가 있다.

또한, 현재는 새로운 이미지에 대한 분석 요청이 들어오면 동시에 분석이 진행되지 않고, 순차적으로 처리되고 있다. 이는 향후 여러 대의 동적분석 서버 구축 및 병렬 처리를 통해 해결하여야 한다

5. 결 론

5.1 결 론

도커 허브 내 도커 이미지를 대상으로 정적분석 및 샌드박스 기반 동적분석을 진행하여 결과 보고서와 조치 가이드를 제공하는 웹 서비스를 개발하였다. 정적분석은 오픈소스 취약점 스캐너인 Trivy를 활용하였으며, 동적분석은 안전한 환경에서 분석이 수행될 수 있도록 샌드박스 환경을 구축하여 진행하였다. 동적분석은 Root 권한 실행, 네트워크, 열린 프로세스, 컨테이너 내 자동 실행되는 파일, 총 4가지 항목을 대상으로 분석을 진행한다. 분석 결과를 바탕으로 결과 보고서를 제공하며, 사용자가 분석 결과에 대응할 수 있도록 조치 가이

드 또한 함께 제공한다. 조치 가이드는 도커 이미지 취약점 분석 평가 항목과 그에 따른 위험도를 표시한 체크리스트, 분석 항목별 진단 방법과 조치 방법을 포함하여 제작하였다.

5.2 기대효과

기존 도커 이미지 분석 도구와 달리 웹으로 제공하기 때문에 사용자가 편리하게 분석 서비스를 이용할 수 있다. 또한, 조치 가이드를 통해 사용자에게 발생할 수 있는 2차 피해를 예방할 수 있을 것으로 기대된다.

6. 별 첨

6.1 팀원 소개

이름	GitHub 주소	수행 파트
유재겸(팀장)	https://github.com/jagym105	샌드박스 구현
서민재	https://github.com/seomj	도커 이미지 정적분석 기능 구현
이다영	https://github.com/young-da	도커 이미지 동적분석 기능 구현
이유경	https://github.com/lyk00331	도커 이미지 진단 조치 가이드 제작
김우종	https://github.com/c0wb3ll	웹 서비스 개발
남서현	https://github.com/SHNam00	웹 서비스 개발
장혜선	https://github.com/haehae00	웹 서비스 개발

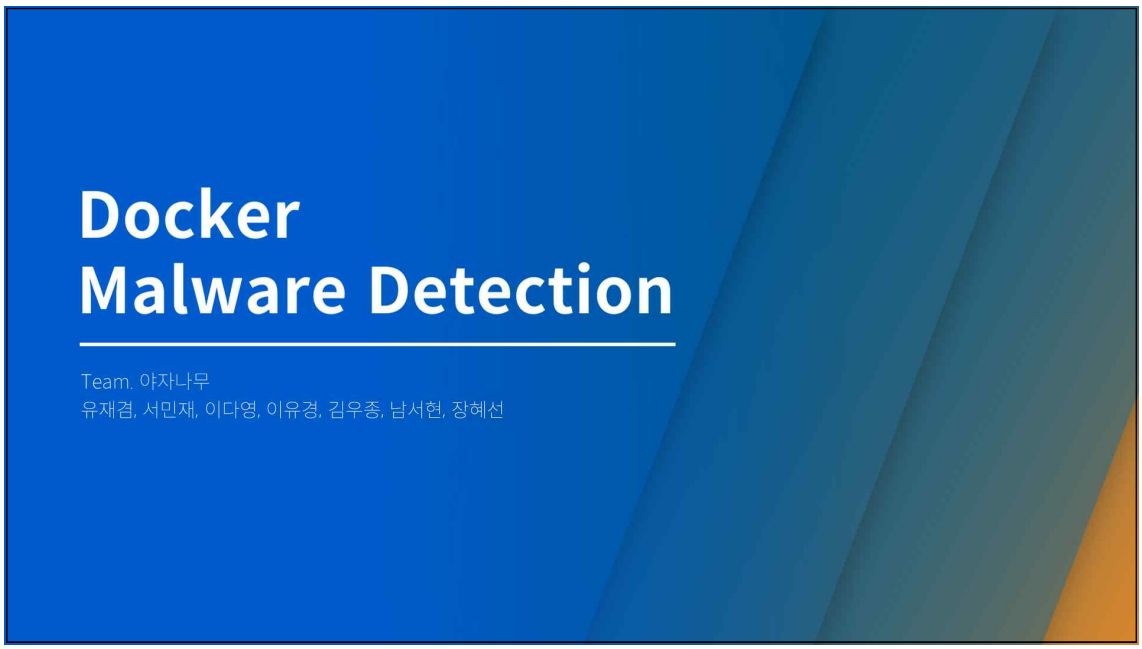
6.2 소스 코드

https://github.com/YazaTree/Docker_Malware_Detect

6.3 시연 영상

https://www.youtube.com/watch?v=97UO16rSicY

6.4 발표 자료


Docker
Malware Detection
Team. 야자나무
유재겸, 서민재, 이다영, 이유경, 김우종, 남서현, 장혜선

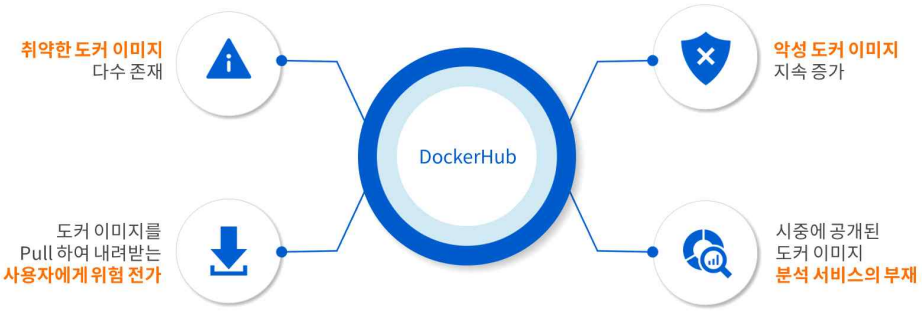

목차
01
프로젝트 소개
1-1. 팀원 소개
1-2. 프로젝트 배경
1-3. 프로젝트 목적
02
프로젝트 개발
2-1. 구상도
2-2. 정적분석
2-3. 샌드박스 및 동적분석
2-4. 웹 사이트
2-5. 컨설팅
03
프로젝트 결과
3-1. 시연 영상
3-2. 결론 및 기대효과
2

1-1. 팀원 소개

이름	수행 파트
유재겸(팀장)	샌드박스 구현
서민재	도커 이미지 정적분석 기능 구현
이다영	도커 이미지 동적분석 기능 구현
이유경	도커 이미지 진단 조치 가이드 제작
김우종	웹 서비스 개발
남서현	웹 서비스 개발
장혜선	웹 서비스 개발

1. 프로젝트 소개

1-2. 프로젝트 배경



1. 프로젝트 소개

1-3. 프로젝트 목적

편리한 분석

웹 서버와 분석 서버 간 연동을 통해
실시간 분석 기능을 제공함으로써
사용자의 **분석 편의성 증대**

악성 도커 이미지 대비

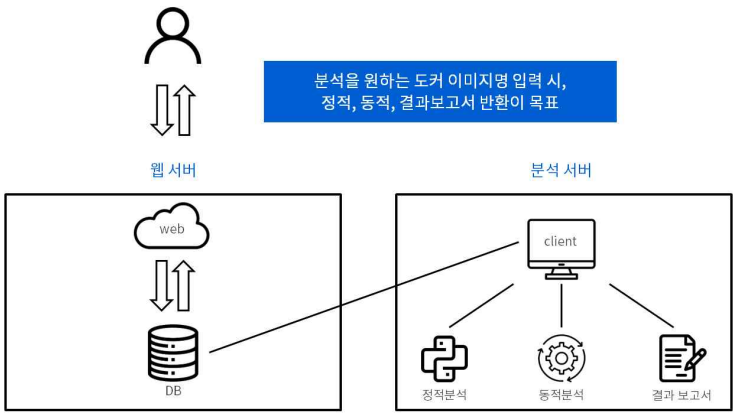
도커 이미지 **정적/동적 분석**을 통해 정확한
탐지 결과를 도출하고,
조치 가이드를 제공하여 **취약한 도커 이미지**
및 악성 도커 이미지에 대비

02

프로젝트 개발

2-1. 구상도
2-2. 정적분석
2-3. 샌드박스 및 동적분석
2-4. 웹 사이트
2-5. 컨설팅

2-1. 구상도



2-2. 정적분석

Trivy를 이용한 CVE 진단

```
centos:latest (centos 8.4.2105)
================================
Total: 439 (UNKNOWN: 0, LOW: 158, MEDIUM: 253, HIGH: 28, CRITICAL: 0)

┌──────────────────┬────────────────┬──────────┬───────────────────┬───────────────────┬──────────────────────────────────────────────────────────────┐
│     Library      │ Vulnerability  │ Severity │ Installed Version │   Fixed Version   │                            Title                             │
├──────────────────┼────────────────┼──────────┼───────────────────┼───────────────────┼──────────────────────────────────────────────────────────────┤
│ bind-export-libs │ CVE-2021-25215 │ HIGH     │ 32:9.11.26-3.el8  │ 32:9.11.26-4.el8_4 │ bind: An assertion check can fail while answering queries    │
│                  │                │          │                   │                   │ for DNAME records...                                         │
│                  │                │          │                   │                   │ https://avd.aquasec.com/nvd/cve-2021-25215                   │
│                  ├────────────────┤          │                   ├───────────────────┼──────────────────────────────────────────────────────────────┤
│                  │ CVE-2022-38177 │          │                   │ 32:9.11.36-3.el8_6.1 │ bind: memory leak in ECDSA DNSSEC verification code        │
│                  │                │          │                   │                   │ https://avd.aquasec.com/nvd/cve-2022-38177                   │
│                  ├────────────────┤          │                   ├───────────────────┼──────────────────────────────────────────────────────────────┤
│                  │ CVE-2022-38178 │          │                   │                   │ bind: memory leaks in EdDSA DNSSEC verification code         │
│                  │                │          │                   │                   │ https://avd.aquasec.com/nvd/cve-2022-38178                   │
│                  ├────────────────┤          ├───────────────────┼───────────────────┼──────────────────────────────────────────────────────────────┤
│                  │ CVE-2021-25214 │ MEDIUM   │                   │ 32:9.11.26-6.el8  │ bind: Broken inbound incremental zone update (IXFR) can      │
│                  │                │          │                   │                   │ cause named to terminate...                                  │
│                  │                │          │                   │                   │ https://avd.aquasec.com/nvd/cve-2021-25214                   │
```

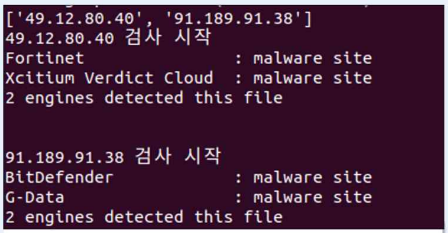
Trivy를 이용하여 Docker image의 **CVE** 정보를 추출하는 Python 자동화 스크립트 작성

2-3. 샌드박스 및 동적분석

샌드박스 동적분석 동작 흐름

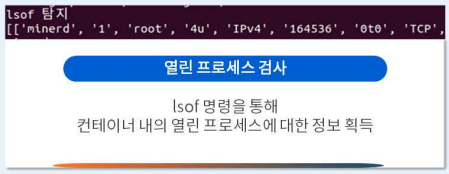
2-3. 샌드박스 및 동적분석

동적분석 과정



네트워크 검사

netstat 명령 수행 후
IP 추출하여 악성 검사 진행



열린 프로세스 검사

lsof 명령을 통해
컨테이너 내의 열린 프로세스에 대한 정보 획득

자동 실행 파일 검사

docker history 명령을 통해 도커 이미지 레이어 추출 후
자동 실행 파일 확인 (ENTRYPOINT)
-> 해당 파일 컨테이너 외부로 복사하여 검사 진행

2-4. 웹 사이트

구조도

메인 페이지

▲ docker image명을 전달받는 페이지

결과 페이지

▲ 분석 결과를 사용자에게 보여주는 페이지

Docker web

- docker_img [분석 대기 리스트]
- static [정적분석 결과]
- dynamic [동적분석 결과]
- fs [결과 보고서]
- recent_search [최근 검색한 도커 이미지]

2-4. 웹 사이트

구조도

사용자가 도커 이미지명 입력

1) DB에 분석 결과 존재 시 해당 결과 출력

2) docker_img에 없을 시 이미지명 저장

3) docker_img에 존재 시 분석 진행

static

주기적 요청 데이터 반환

2-4. 웹 사이트

메인 페이지

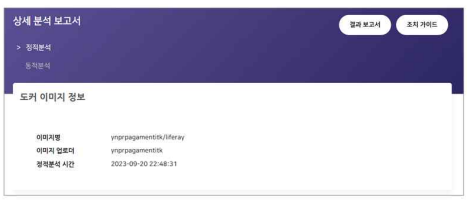
도커 이미지 검색창

최근 검색한 도커 이미지 리스트

2-4. 웹 사이트

정적분석 결과 페이지

도커 이미지 정보



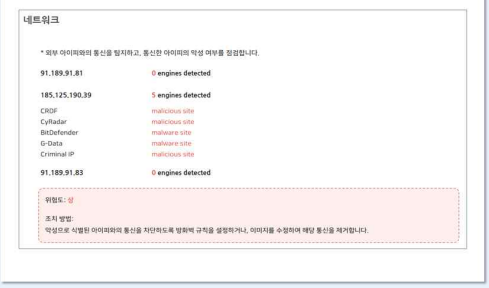
보안 취약점 (CVE) 검사

2-4. 웹 사이트

동적분석 결과 페이지

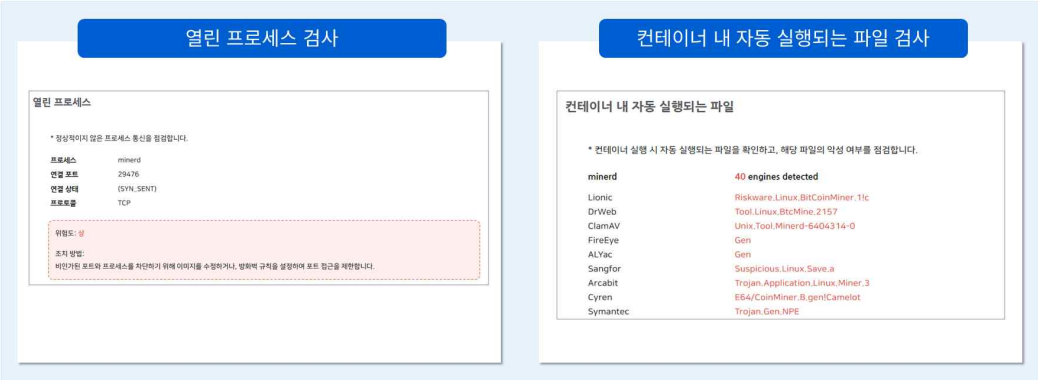
Root 실행 여부 검사

네트워크 검사



2-4. 웹 사이트

동적분석 결과 페이지



2-5. 컨설팅

수행 절차

2-5. 컨설팅

결과 보고서 및 조치 가이드

결과 보고서

도커 이미지 취약성 결과 보고서

스캔 정보

Image name: ynprpagamentos/liferay
Start time: 2023-09-20 22:48:31
Finish time: 2023-09-20 22:50:55
Elapsed: 143sec

수행된 항목 목록

구분	진단 항목	개수/결과
CVE 진단	CVE Critical	1
	CVE High	63
	CVE Medium	716
	CVE Low	412
	CVE Unknown	0
권한 설정	Root 계정	탐지
접근 통제	외부 아이피 통신	탐지
	TCP 통신 프로세스	탐지
실행 파일 탐지	자동 실행 파일	탐지

자동화 진행하여 분석 종료 시 보고서 자동 반환

체크리스트 (조치 가이드中)

1. 체크리스트 항목

도커 이미지 취약점 진단에 사용될 체크리스트는 국내외 기술 자료를 바탕으로 작성하였다.
가이드 내 영역은 크게 정적 분석, 동적 분석으로 구성하였으며, 정적 분석은 CVE 진단(5개 항목), 동적 분석은 권한 설정(1개 항목), 접근 통제(2개 항목), 리소스 관리(1개 항목)로 총 9개 항목으로 제작하였다.

구분	진단 코드	진단 항목	위험도
가. CVE 진단	DS 1-1	오래된 소프트웨어 구성	N/A
	DS 1-2	안전하지 않은 코딩	N/A
	DS 1-3	네트워크 보안	N/A
	DS 1-4	잘못 구성된 설정	N/A
	DS 1-5	악한 인증 및 액세스 제어	N/A
나. 권한 설정	DD 1-1-1	Root 계정: 탐지	중
	DD 1-1-2	Root 계정: 이탈지	N/A
다. 접근 통제	DD 2-1-1	외부 아이피 통신: 악성 아이피로 분류	상
	DD 2-1-2	외부 아이피 통신: 악성 아이피로 이분류	중
	DD 2-1-3	외부 아이피 통신: 이탈지	하
	DD 2-2-1	TCP 통신 프로세스: 탐지	상
	DD 2-2-2	TCP 통신 프로세스: 이탈지	하
라. 실행 파일 탐지	DD 3-1-1	자동 실행 파일: 악성코드로 분류	상
	DD 3-1-2	자동 실행 파일: 악성코드로 이분류	중
	DD 3-1-3	자동 실행 파일: 이탈지	하

[표 2] 도커 이미지 취약점 분석평가 항목

조치 가이드

2.2. 네트워크 검사

DD 2-1-1	다. 접근 통제 > 외부 아이피 통신: 악성 아이피로 분류	위험도	상
취약점 개요			
점검 내용	■ 외부 아이피와의 통신 탐지 및 통신한 아이피의 악성 여부 점검		
점검 목적	■ 악성 아이피 통신으로 인한 악성코드 다운로드 등 문제를 방지하기 위함		
보안 위협	■ 시스템 해킹, 악성코드 전파, 개인정보 유출 등의 문제를 조래할 수 있음		
진단 기준 및 진단 방법			
진단 기준	■ 외부 아이피와 통신 및 해당 아이피가 악성으로 식별되는 경우		
진단 방법	■ 아래 명령어를 통해 네트워크 연결 상태 확인 및 현재 통신하는 아이피 확인 # netstat -an ■ 검출된 외부 아이피에 대해 VirusTotal을 이용하여 진단 후, 악성 아이피로 확인됨		
조치 방법			
조치 방법	■ 악성으로 식별된 아이피와의 통신을 차단하도록 방화벽 규칙을 설정하거나, 이미지를 수정하여 예상 통신을 제거함		

[표 11] 세부 항목 설명

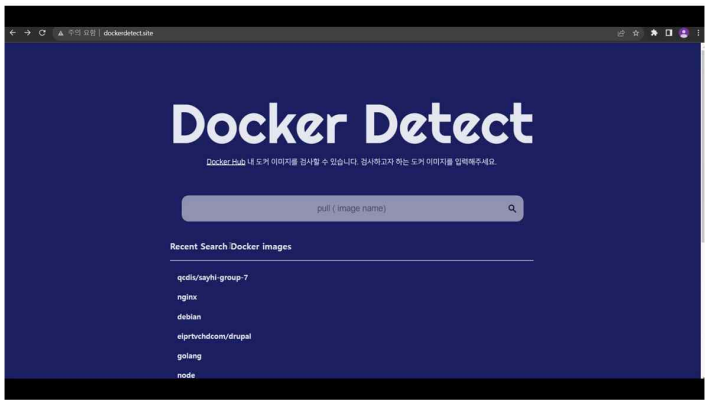
03

프로젝트 결과

3-1. 시연 영상

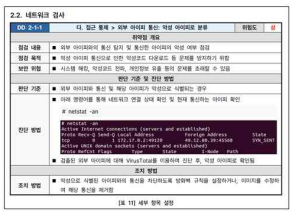
3-2. 결론 및 기대효과

3-1. 시연 영상



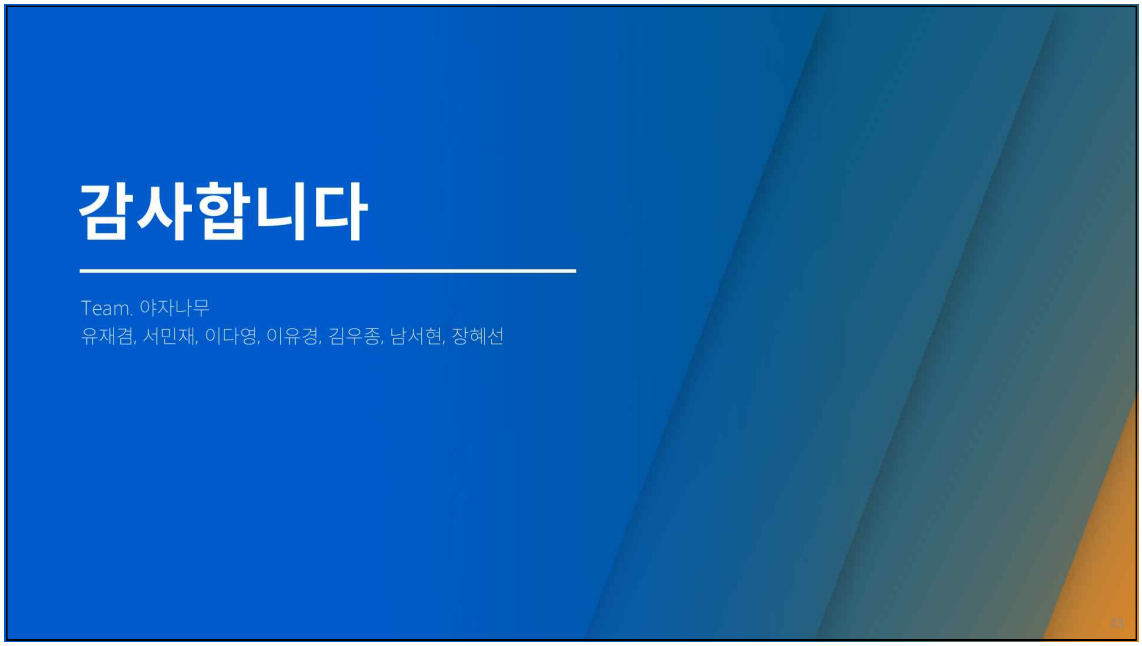
시연 영상 주소: https://youtu.be/97UO16rSicY | 웹 서비스 주소: http://dockerdetect.site

3-2. 결론 및 기대효과



도커 이미지 분석 서비스 개발 및 조치 가이드 제작

1. 웹을 통해 분석 서비스를 제공함으로써 분석 용이성 증대
2. 샌드박스를 활용한 도커 이미지 동적분석 방안 연구
3. 조치 가이드를 통해 사용자에게 발생할 수 있는 2차 피해 예방


감사합니다
Team. 야자나무
유재겸, 서민재, 이다영, 이유경, 김우종, 남서현, 장혜선

6.5 소개 자료

2023 정보보호학과 졸업작품 전시회

Docker Malware Detection

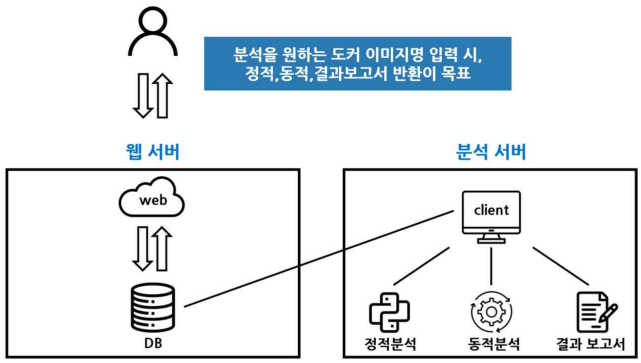
개요

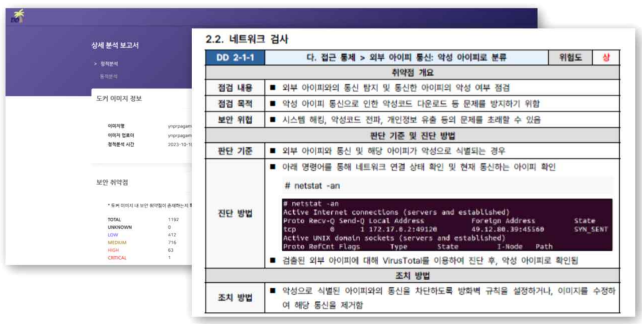
도커허브 내 다수의 도커 이미지에는 여러 취약점이 존재하며, 악성 도커 이미지 또한 증가하고 있다. 이로 인한 피해 예방을 목표로 악성 도커 이미지 정적·동적 분석을 진행하고, 결과 보고서 및 조치 가이드를 제공하는 웹 서비스를 개발하였다.

팀 소개

지도교수	이병천 교수님
유재겸	샌드박스 기능 구현
서민재	도커 이미지 정적분석 기능 구현
이다영	도커 이미지 동적분석 기능 구현
이유경	도커 이미지 진단 조치 가이드 제작
김우종	웹 서비스 개발
남서현	웹 서비스 개발
장혜선	웹 서비스 개발

서비스 구성도 및 결과물





도커 이미지 분석 서비스 및 조치 가이드

Yazatree