[DFC-2021 104] Write-Up

작성자	박혜미
분석 일자	2024.05.29~2024.06.09
작성 일자	2024.06.09
분석 대상	alice_Memory.dd alice_virtualMachine.E01 output.zip
문서 버전	1.0
작성자 E-mail	parkm0708@naver.com

0. 목차

1. 문제

URL	-
문제 내용	Alice set up development environment for the development of the service module related with membership. Here is the virtual machine where Alice has built the development environment. (문제에 대한 모든 내용은 [6. 별도 첨부]한다.
문제 파일	(문제 파일 용량이 너무 큰 관계로 삽입하지 않는다.)
문제 유형	system forensics
난이도	3 / 3

2. 분석 도구

도구명	다운로드 링크	Version
OSFMount	https://www.osforensics.com/tools/mount-disk-images.html	3.1.1003.0
FTK Imager	https://www.exterro.com/digital-forensics-software/ftk-imager	4.7.1.2
VScode	https://code.visualstudio.com/download	1.90.0

3. 환경

OS
Windows 11 Home

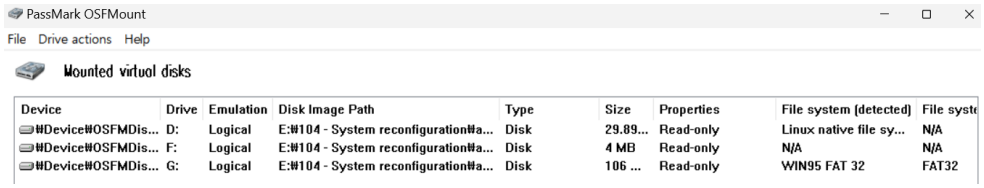
4. Write-Up

파일명	alice_Memory.dd
용량	10.3GB
SHA256	f673370a57bf7e8281c699eb83812ba3d3e1e8bd7855c0b6d77244c43dfbf7fc
Timestamp	2021-06-22 21:01:48

파일명	alice_virtualMachine.E01
용량	5.06GB
SHA256	3fd0419bdce85ba975c5803829a5ecc3d7f813b2b468058068b70dc63e243fb4
Timestamp	2021-06-23 17:31:34

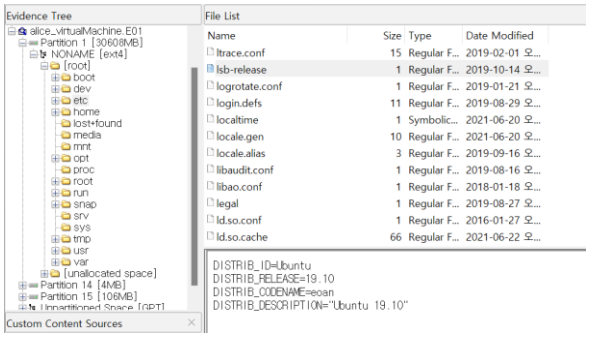
파일명	output.zip
용량	1.18MB
SHA256	d9a364ff8d4410a0faa488efb969d94e51ca49c2baba3dff6658323ab4ce6d1f
Timestamp	2021-06-30 10:18:30

1. 도커를 분석하여 다음 정보를 확인합니다.



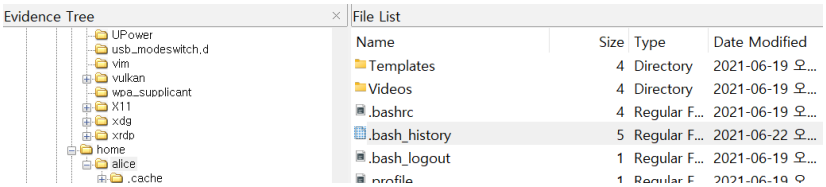
[그림 1] alice_virtualMachine.E01 마운트 한 모습

OSFMount 프로그램을 사용하여 alice_virtualMachine.E01 파일을 마운트 해준다.



[그림 2] OS 발견

FTK Imager 를 사용하여 alice_virtualMachine.E01 을 열어보았더니 /etc/lsb-release 에서 OS 를 알아낼 수 있었다. 문제 파일의 OS 는 Ubuntu 19.10 이다.



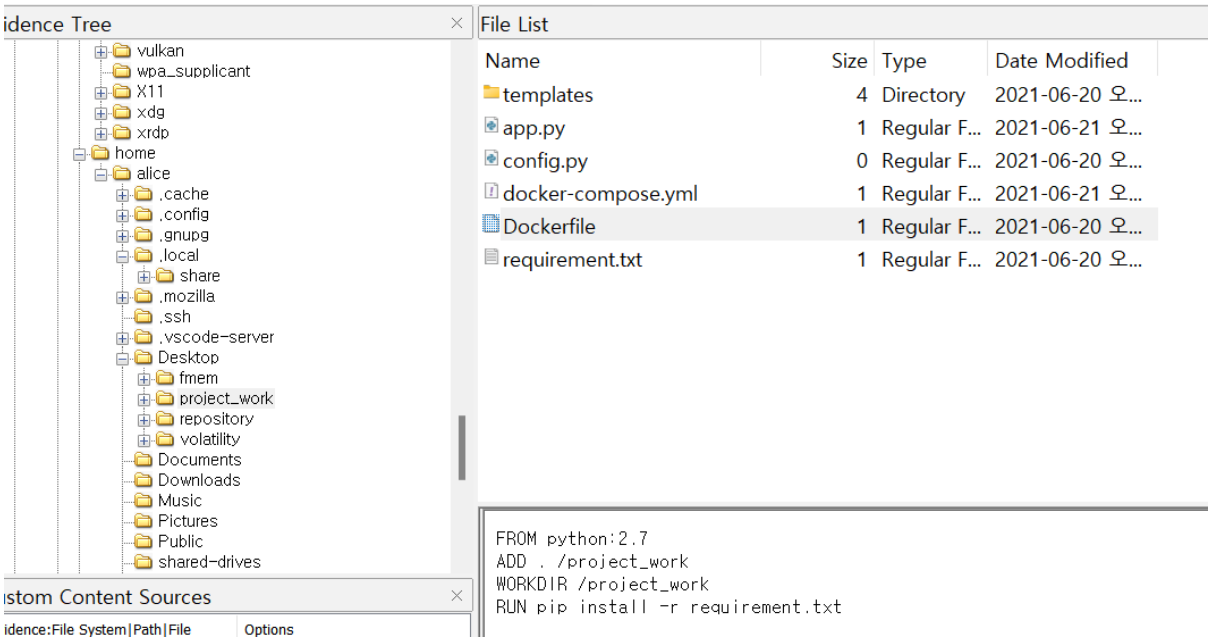
[그림 3] /home/alice/.bash_history

파티션 1 에 alice 폴더가 존재한다. 해당 폴더 안에 '.bash_history'라는 파일이 있는 것을 보아 bash 를 사용하는 유저인 듯하다. 해당 파일을 분석해 보겠다.

```
ls -al
vim ~/.git
cat ./*.git
cat ./.git/*
cd ..
cd ./project_work/
touch app.py
touch config.py
mkdir templates
touch templates/project_work.html
touch Dockerfile
touch docker-compose.yml
touch requirement.txt
sudo vim ./app.py
sudo vim requirement.txt
sudo vim ./Dockerfile
sudo vim ./docker-compose.yml
sudo docker image
sudo docker images
sudo vim ./docker-compose.yml
cd ./templates/
sudo vim ./project_work.html
cd ..
sudo service docker start
sudo docker-compose up
vim ./docker-compose.yml
sudo docker-compose up
docker pull mongo:2.6.2
sudo docker pull mongo:2.6.2
```

[그림 4] .bash_history 중 한 부분

해당 파일은 bash 쉘 사용 기록이다. 내용을 보니 'project_work' 폴더를 주로 사용하였다. 해당 폴더에서 'docker-compose.yml' 파일을 작성하고 docker compose 를 사용해 이미지를 구성하였다.

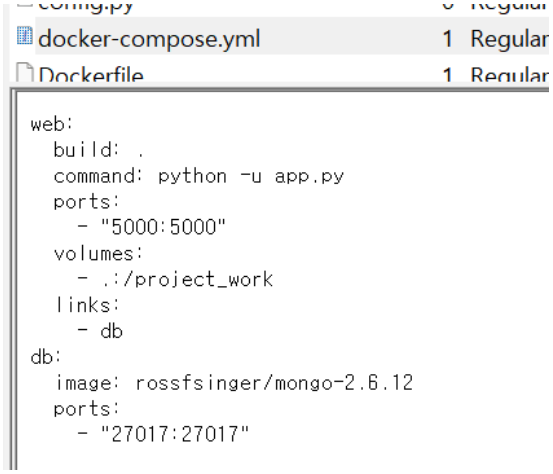


[그림 5] alice/Desktop/project_work/Dockerfile

alice/Desktop/project_work/Dockerfile 을 살펴보니 python 을 사용하였으며, requirement.txt 을 작동하였다.

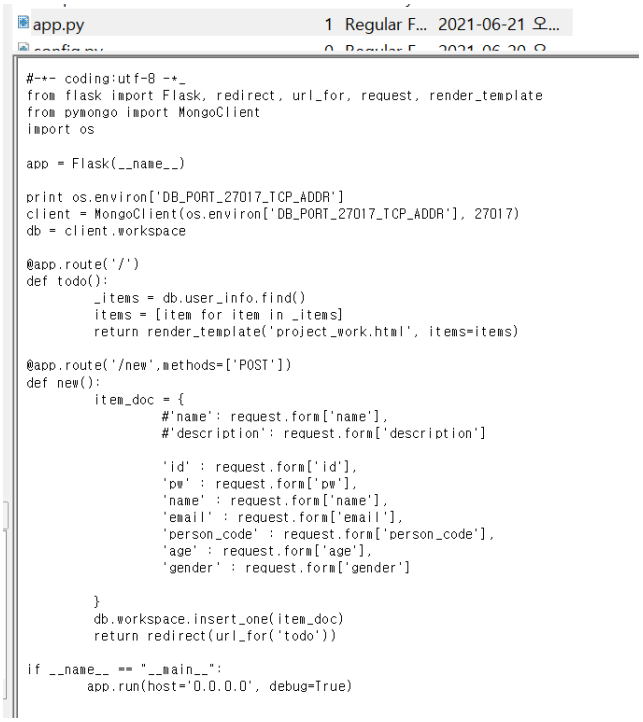
[그림 6] alice/Desktop/project_work/requirement.txt

requirement.txt 안의 내용을 살펴보니 사용자가 작동시킨 것은 파이썬 패키지인 flask 와 pymongo 을 설치한 것 같다.



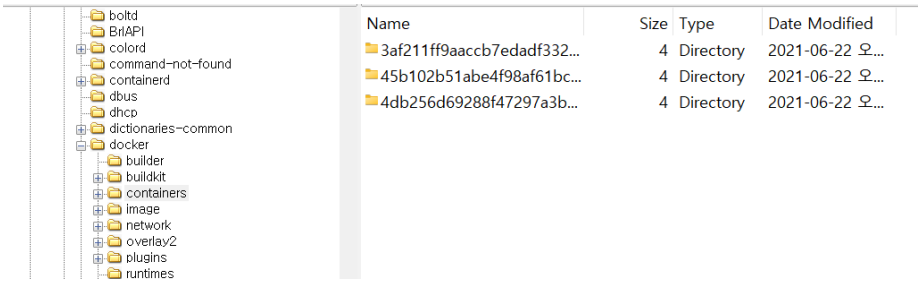
[그림 7] alice/Desktop/project_work/Docker-compose.yml

alice/Desktop/project_work/Docker-compose.yml 을 확인해 보니, docker-compose 를 사용하면 해당 컨테이너에서 'python -u app.py'을 실행하고, 5000 번 포트를 사용했다. 또한 DB 를 연결할 때 rossfsinger/mongo-2.6.12 을 사용하여 27017 포트를 사용했다.



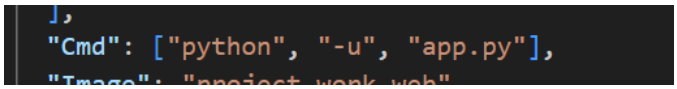
[그림 8] alice/Desktop/project_work/app.py

app.py 는 모듈을 만든 코드이다. Alice 가 개발했다는 멤버십 관련 서비스 모델은 해당 코드를 뜻하는 것 같다.



[그림 9] /var/lib/docker/containers

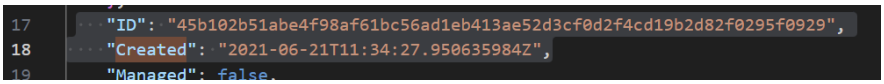
컨테이너 로그를 찾기 위해 /var/lib/docker/containers 폴더로 이동한다. 확인해 보니 총 3 개의 컨테이너가 있었다. 이중 alice 가 개발한 멤버십 앱의 컨테이너는 45 로 시작하는 컨테이너이다.



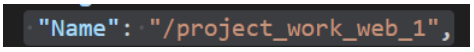
[그림 10] config.v2.log(45) 중 일부분 1

[그림 11] config.v2.log(45) 중 일부분 2

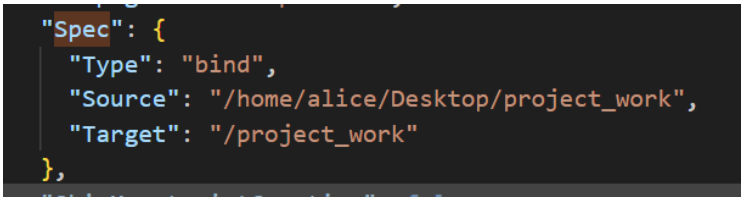
해당 컨테이너에서 config.v2.log 을 추출하여 VScode 를 사용하여 열어보고 살펴보았더니 **python** -u app.py 명령어로 실행되고, **TCP 5000 번 포트**를 사용하는 것을 확인할 수 있었다.



[그림 12] config.v2.log(45) 중 일부분 3



[그림 13] config.v2.log(45) 중 일부분 4



[그림 14] config.v2.log(45) 중 일부분 5

생성시간은 **2021-06-21 11:34:27** 이다.

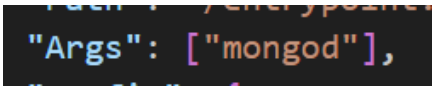
컨테이너 ID 는

45b102b51abe4f98af61bc56ad1eb413ae52d3cf0d2f4cd19b2d82f0295f0929 이다.

이름은 **project_work_web_1** 이다. /home/alice/Desktop/project_work 폴더를 project_work 에 bind 타입으로 마운트한 것을 살펴볼 수 있었다.

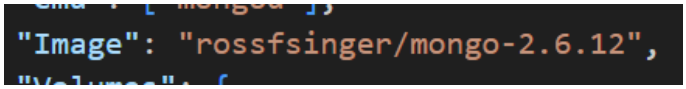
[그림 15] config.v2.log(45) 중 일부분 6

2021-06-21 11:48:56 에 시작되었고, 2021-06-22 12:02:17 에 종료된 것을 볼 수 있다.

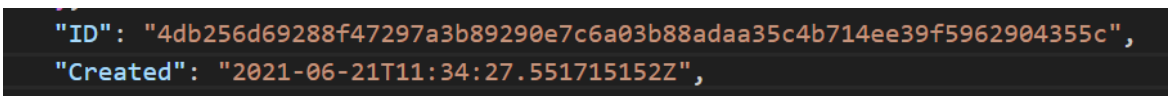


[그림 16] config.v2.log(4db) 중 일부분 1

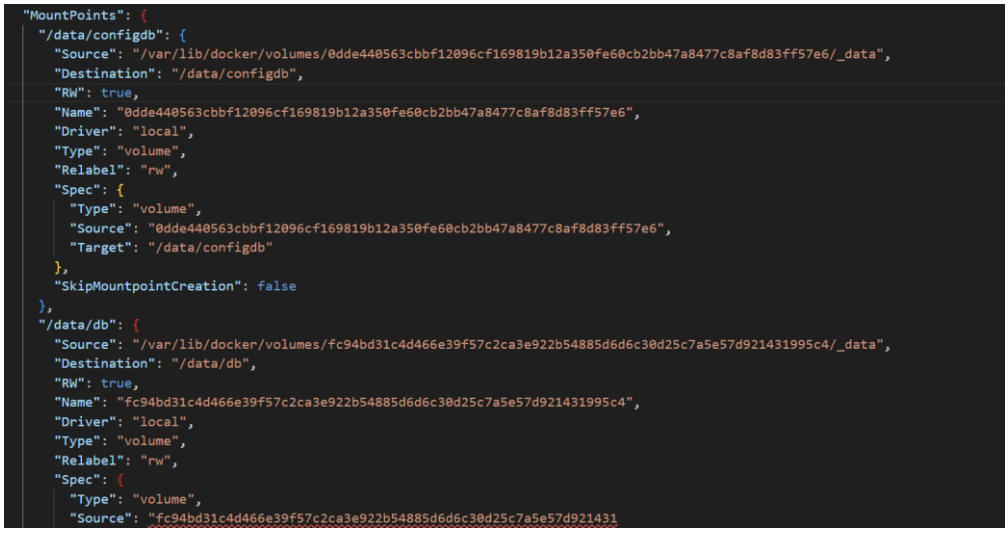
[그림 17] config.v2.log(4db) 중 일부분 2



[그림 18] config.v2.log(4db) 중 일부분 3



[그림 19] config.v2.log(4db) 중 일부분 4



[그림 20] config.v2.log(4db) 중 일부분 5

[그림 16] 부터는 mongodb 컨테이너인 4db 로 시작되는 컨테이너다. 해당 컨테이너 또한 config.v2.log 을 추출하여 VScode 를 사용해 분석해 보았다. **Rossfsinger/mongo-2.6.12** 를 사용하고, 2021-06-21 11:34:27 에 생성됐다.

ID 는 **4db256d69288f47297a3b89290e7c6a03b88adaa35c4b714ee39f5962904355c** 이며, 컨테이너 이름은 **project_work_db_1** 이다. Volume 타입으로 local 디렉터리가 마운트 되는 것은 아래와 같다.

Local 디렉터리	마운트 되는 것
/var/lib/docker/volumes/0dde440563cbbf12096cf169819b12a350fe60cb2bb47a8477c8af8d83ff57e6/_data	/data/configdb
/var/lib/docker/volumes/fc94bd31c4d466e39f57c2ca3e922b54885d6d6c30d25c7a5e57d921431995c4/_data	/data/db

또한, **2021-06-21 11:48:55 에 실행되어 2021-06-22 12:02:17 에 종료**되었으며, 멤버십 앱 컨테이너도 동시에 실행되고, 종료되는 것을 확인할 수 있다.

네트워크 연결 정보도 파악해야 하기 때문에 메모리 이미지 또한 분석하였다. Volatility 를 사용하기 위해 OS profile 정보가 필요한데, 현재 Volatility 는 Ubuntu 19.10 버전의 profile 정보를 제공하고 있지 않다. 따라서, 해당 문제를 푸는 데엔 어려움이 있다.

5. Flag

1.도커를 분석하여 다음 정보를 확인합니다.

¬ 컨테이너 ID & 이름

컨테이너 ID	Name
45b102b51abe4f98af61bc56ad1eb413ae52d3cf0d2f4cd19b2d82f0295f0929	project_work_web_1
4db256d69288f47297a3b89290e7c6a03b88adaa35c4b714ee39f5962904355c	project_work_db_1

¬ 컨테이너의 IP & 포트

project_work_web_1 : 172.18.0.3:5000

project_work_db_1 : 172.18.0.2:27017

¬ 컨테이너에 있는 응용 프로그램 서비스 버전입니다.

project_work_web_1 : Python 2.7.18

project_work_db_1 : MongoD 2.6.12

¬ 컨테이너의 생성, 시작 및 종료 시간.

컨테이너	생성 시간	시작 시간	종료 시간
project_work_web_1	2021-06-21 11:34:27	2021-06-21 11:48:56	2021-06-22 12:02:17
project_work_db_1	2021-06-21 11:34:27	2021-06-21 11:48:55	2021-06-22 12:02:17

2.도커를 이용하여 구성한 서비스 중에서,

¬ 인증에 취약한 응용 프로그램을 찾습니다.

¬ 앱을 취약한 것으로 결정한 이유를 설명합니다.

3.취약성을 통해 유출될 수 있는 데이터는 무엇입니까?

¬ 파일 이름

¬ 해시값(MD5)

6. 별도 첨부

- 문제 번역본

Alice 는 멤버십 관련 서비스 모듈 개발을 위한 개발 환경을 마련하였습니다. Alice 가 개발 환경을 구축한 가상 머신입니다.

1.도커를 분석하여 다음 정보를 확인합니다.
¬ 컨테이너 ID & 이름
¬ 컨테이너의 IP & 포트
¬ 컨테이너에 있는 응용 프로그램 서비스 버전입니다.
¬ 컨테이너의 생성, 시작 및 종료 시간.

2.도커를 이용하여 구성한 서비스 중에서,
¬ 인증에 취약한 응용 프로그램을 찾습니다.
¬ 앱을 취약한 것으로 결정한 이유를 설명합니다.

3.취약성을 통해 유출될 수 있는 데이터는 무엇입니까?
¬ 파일 이름
¬ 해시값(MD5)

7. Reference

- [URL]