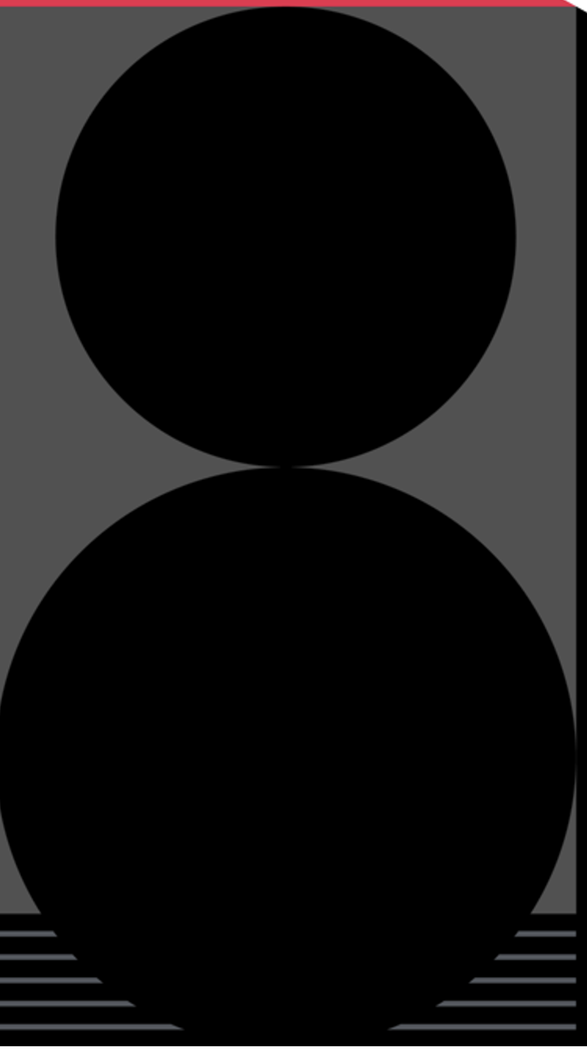

8 Docker 트러블 슈팅 방법

03 네트워크 정보 확인 및 트래픽 Dump/디버깅 방법

실습 진행사항

네트워크 정보 확인 및 트래픽 Dump/디버깅 방법 실습

1. Docker 컨테이너가 사용하고 있는 특정 네트워크 인터페이스를 확인
2. 인터페이스에서 발생하는 트래픽 정보(패킷)등을 tcpdump 파일에 수집
3. Wireshark에 tcpdump 파일 import 및 Debugging 수행

사전 준비사항

- Docker가 설치된 AWS EC2 VM Instance 1개(Ubuntu 18.04 리눅스)
  - 이전 강의 때 생성한 Common VM 사용
- 로컬 개발환경에 Wireshark 설치

Wireshark 설치 방법

설치 파일 다운로드 URL

- https://www.wireshark.org/download.html

macOS 설치

- macOS Intel 64-bit .dmg 다운로드 후 설치

Windows 설치

- Windows Installer (64-bit) 다운로드 후 설치

실습1. Docker 컨테이너가 사용하고 있는 특정 네트워크 인터페이스를 확인 방법

특정 Docker 컨테이너 상세 정보 확인 명령어

- docker inspect <실행중인 컨테이너 ID>

Docker 컨테이너가 사용하고 있는 특정 네트워크 인터페이스 확인 명령어

- ip addr

실습2. 인터페이스에서 발생하는 트래픽 정보등을 tcpdump 파일에 수집 방법

tcpdump 수집 명령어

- tcpdump –i <컨테이너 네트워크 인터페이스명> -w <Dump 파일명>

tcpdump 파일 로컬로 가져오는 방법

- scp –i <서버접속 SSH Key> <계정명>@<서버 IP 혹은 DNS 주소>:<Dump 파일이 있는 서버의 경로> <Dump 파일을 저장할 로컬환경의 경로>

실습3. Wireshark에 tcpdump 파일 import 및 Debugging 수행 방법

Wireshark Import 방법

- 파일 > import > scp로 가져온 Dump 파일 선택 > 열기

Wireshark Debugging 방법

- filter 창에서 다음의 키워드로 검색

- http
- tcp.dstport == 80
- tcp.port == 80 and ip.addr == <특정 목적지 IP>
- http.request.method == “GET”
- http.response.code == 200