Isolation

Dileepa Fernando

Isolation

- OS Isolation –

https://pdos.csail.mit.edu/6.828/2008/lec/l-interrupt.html (INT instruction)
https://github.com/torvalds/linux/blob/master/arch/x86/entry/entry_32.S
(Interrupt handler in Kernel)

https://meltdownattack.com/meltdown.pdf (Meltdown Attack)

- VM Isolation
- Container Isolation

Isolation

- You need a VM running linux
- Docker Installation and Features
  - Pre installation setup
  - Installation
  - Use cases
- Follow the linux commands and execute
  - Guess the semantics of each command
  - Explore the docs to confirm

Pre Installation (for debian linux)

- *sudo apt update*
- *sudo apt install apt-transport-https ca-certificates curl gnupg lsb-release*

- *curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg*
- *echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null*
- *sudo nano /etc/apt/sources.list.d/docker.list* (Verify the above is correctly written)

Installation

- Install
  - *sudo apt update*
  - *apt-cache policy docker-ce*
  - *sudo apt install docker-ce docker-ce-cli containerd.io*

- Verify docker is running
  - *sudo systemctl status docker*

- User configuration (After this you do not need sudo for docker operations)
  - *sudo usermod -aG docker ${USER}*
  - *su - ${USER}*
  - *id -nG*

  Restart the machine

Use Cases

- Download two docker images
  - *docker pull ubuntu:latest*
  - *docker pull nginx:latest*
- List down the current docker images
  - *docker images*
- Run a container in another terminal
  - *docker run --rm -it ubuntu:latest /bin/bash*
- Try running terminal commands inside container

Use Cases

- Run a process (Ex: *sleep 30*) in a container and view PID
  - Within the container (Can use *&*)
  - From the VM hosting the container (Can use *ps* and *grep* commands)

- Run the same process (Ex: *sleep 30*) within chroot and view PID
  - Within newroot
  - Outside of newroot

Use Cases

- Run two containers in two terminals
  - Check the pid of the terminals within the container
    - *echo $$*
  - Check from the pids from the host
    - *docker inspect …*

Build your own container with python

- *sudo docker build --rm -t mypy .*
- Execute the above
- Why do you have an error?

- You need to write a Dockerfile with instructions
  - Check this https://docs.docker.com/get-started/02_our_app/
    - Start from a base docker image (preferably python interpreter image)
    - Define working directory of container (you can choose one)
    - Copy the files from the host to the container
    - Define the command to execute when container starts (python your-script.py)
    - Use the shared python script from gitlab Isolation Folder
    - You may edit the shared Dockerfile
- Re run the build command now

Run your containerized python application

- *docker run -it --rm mypy*
- Inspect the output

- Try running it by defining 100MB memory limit
- Inspect the output

- Inspect the memory usage of the container from a different terminal
  - *docker stats*

Syscall Filtering

- Need to define the blacklist and white list of system calls in a json file
- When running the container add option
  - --security-opt seccomp=profile.json
- *sudo docker run -it --rm --security-opt seccomp=profile.json ubuntu /bin/bash*
- In the new terminal
  - Create a text file
  - Try to perform chmod
  - Inspect the result

Additional Technologies

- Kubernetes
  - Managing containers across different hosts
  - Scaling cloud applications