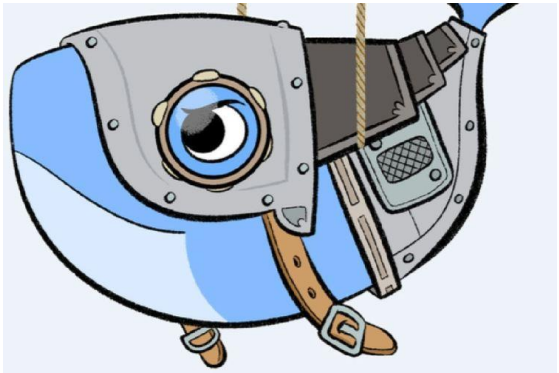
Security in Docker Containers

Aristotelis Kotsias
Georgios Tagkoulis

Why Docker security



Massive use of containers technology from enterprises and organizations result in the introduction of a new set of security considerations

Agenda

- Security features of Docker containers
- Vulnerabilities in Docker containers
- Vulnerabilities Prevention
- Take home message

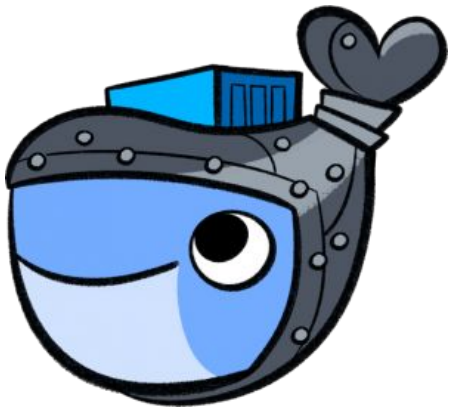
Docker - “Secure by default” approach

Follows the “*Principle of least privilege*” by:

- Providing isolation; host-application and application-application
- Reducing vulnerable host surface area.

So Docker containers provide **application sandboxing** and **resource constraints** using features like namespaces and control groups from Linux.

Container format - libcontainer



- **Namespaces**
  - **pid**: Process isolation
  - **net**: Managing network interfaces
  - **ipc**: Managing access to IPC resources
  - **mnt**: Managing file system mount points
  - **uts:** Isolating kernel and version identifiers
- **Control groups** (cgroups)
  Kernel level functionality that lets Docker control which resources each container can access
- **UnionFS**

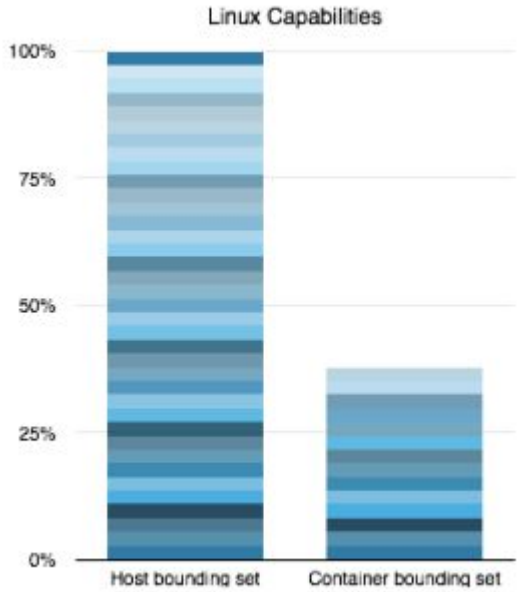
Additional Docker security features

Seccomp (Secure Computing Mode)

Restricts the access that the application container has to the host system to perform actions
https://github.com/moby/moby/blob/master/profiles/seccomp/default.json

(Linux) capabilities

Used to lock down root in a container. They are distinct units of privilege that can be independently enabled or disabled



Vulnerabilities

Malicious open-source images

- Non-authenticated images -> running arbitrary code
- Download images from trusted registry (Docker hub)
- Enable mandatory signature verification (Docker Content Trust)

```
$ export DOCKER_CONTENT_TRUST=1
```

Static docker images

- Running images with outdated libraries or packages
- Use vulnerability scanners (Clair)
- Update and rebuild the image

Source: https://blog.skyboxsecurity.com/docker-vulnerability/

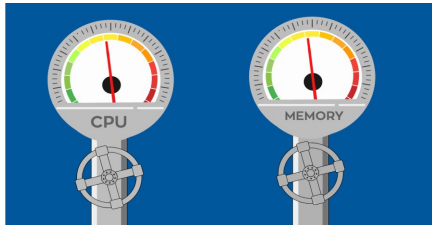
Docker container breakout

- Container escape and access to sensitive information
- Avoid running privileged containers from unknown sources
- Limit the privileges of the container
- Drop unnecessary capabilities

Container resource abuse

- Software bugs or malware attacks that can lead to DoS attacks
- Limit resource usage
- Ensure that containers have only the resources they need

```
$ docker run -it --memory=2G --memory-swap=3G ubuntu bash
```



Take home message

- ➔ Docker although it is “Secure by default”, it is highly configurable
- ➔ Keep up with state-of-the-art security practices