

The Container Security in Healthcare Data Exchange System

Bachelor's degree graduation project

Chih-Hsuan Yang

National Sun Yat-sen University

Advisor: Chun-I Fan

July 9, 2021

Outline

1 Preliminaries

2 Risks

3 Coupled

Preliminaries

What focus on?

- Container security
 - Host environment
 - Images Signature
 - Container behavior
 - Continuous Integration / Continuous Deployment
- Database security
 - Access control
 - Encryption
 - Integration
 - Backups

Aimed security issues

Secure for what?

- CI/CD (Rolling update)
- Strongly access control (Namespaces)
- Reduce leakage possibility (Namespaces)
- Limited resources (Hooked glibc, CGroups, Capabilities)
- Malicious flow detection (LSM)
- E2EE ([Curve25519](#), [chacha20-poly1305](#))

Risks

Comparison

Possible risks without this project's protection

- Container

- Malicious images
- Inner-container permission
- Container Escalations (RCE)
- Out-of-date software
- Infrastructure vulnerabilities

- Database

- Injections
- Malicious flow
- Ownership
 - Broken authentication
 - Encryption failure

Coupled

Stories

Prof. Fan asked: "Is the container and the healthcare data exchange system coupled?"

- ① Can it be coupled, just like the blockchain and cryptography?
 - No. $P \rightarrow Q$
- ② How to make these two issues be coupled?
 - How does the cryptography embedded in blockchain?
 - Authentication process.
 - Find the closest part to embed or entangle.
 - Projection matrix.

The level of matrix

Low (0) means those are orthogonal.

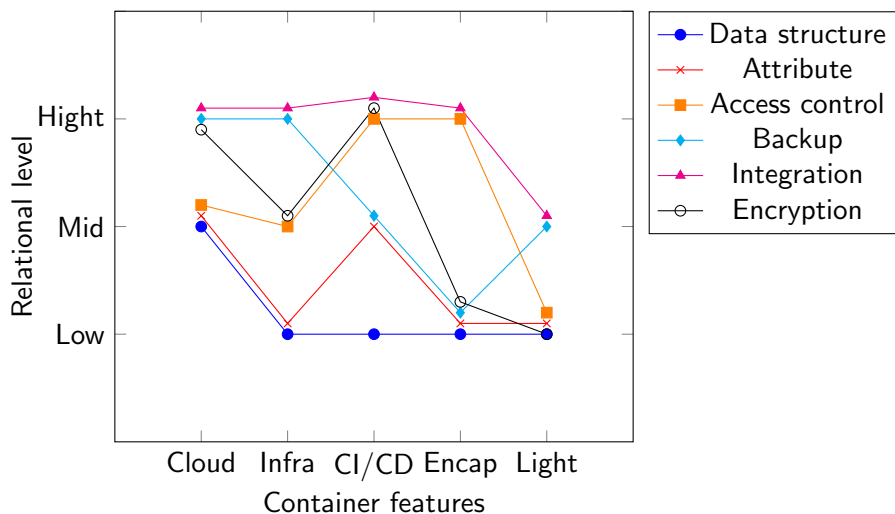
Mid (1) means those are related in some premise.

Hight (2) means strongly bidirectional relation.

Matrix

Relational level between features [0,2]					
	Cloud computing	Infrastructure	CI/CD	Encapsulation	Light weight
Data structure	1	0	0	0	0
Attribute	1	0	1	0	0
Access control	1	1	2	2	0
Backup	2	2	1	0	1
Integration	2	2	2	2	1
Encryption	2	1	0	0	0

Graph



The reason of matrix

- Data structure: Have some relation in the performance issue.
- Attribute (category): Separate into different containers, dynamic CRUD.
- Access control: Could be included in infrastructure's routing rule (LSM).
- Backup: Reliable (but not saved in long term).
- Integration: Separated database needs to integration those caches, rules.
- Encryption: Keep your privacy.

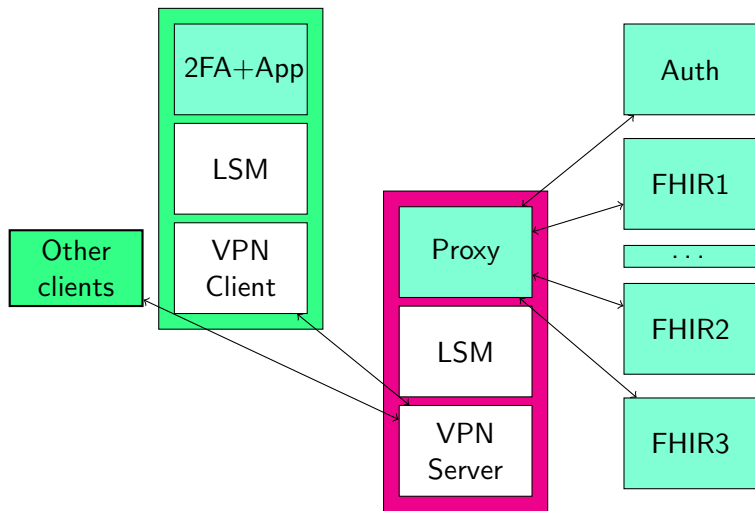
Why we need the container?

The 2FA can solved the robot traffic.

Back to the key advantage of the container:

- Increased portability.
- More consistent operation.
- CI/CD.
 - We can deploy those updates continuously.
- Encapsulation.

Architecture



Pros and cons

- Pros:

- Solved the malicious flow.
- Truly people authorization.
- Coupled with Taiwan healthcare system.

- Cons:

- Modified kernel.
- Slower.
- Coupled with OS.

With the container security issue

- The trend of micro-service on cloud computing.
- CI/CD.
- Encapsulation.

Still not solved:

- Hooked glibc.
- Composed system calls (LSM).

Reference

Wireguard kernel module