The Container Security in Healthcare Data Exchange System

Bachelor's degree graduation project

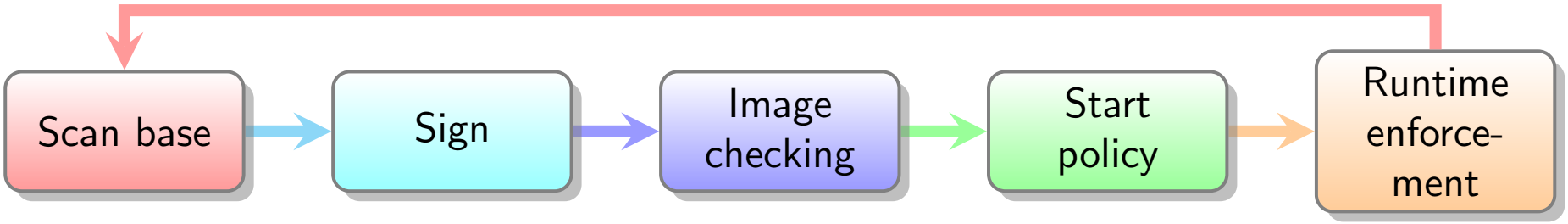
Chih-Hsuan Yang

National Sun Yat-sen University
Advisor: Chun-I Fan

July 30, 2021

Outline

Flow chart


Scan base
Sign
Image checking
Start policy
Runtime enforce-ment

Divide and Conquer

3 steps

1. Sign and check
2. Collect "Normal" samples
3. Enforce policy

Is it can work?

- ✓ Sign and check
  - docker scan
  - docker trust
- ? Collect "Normal" samples
- ✓ Enforce policy in seccomp
  - Wrapper for seccomp.

seccomp - Secure Computing

Not use the LSM.

The seccomp and system call limitation are coupled.

SECCOMP_SET_MODE_FILTER

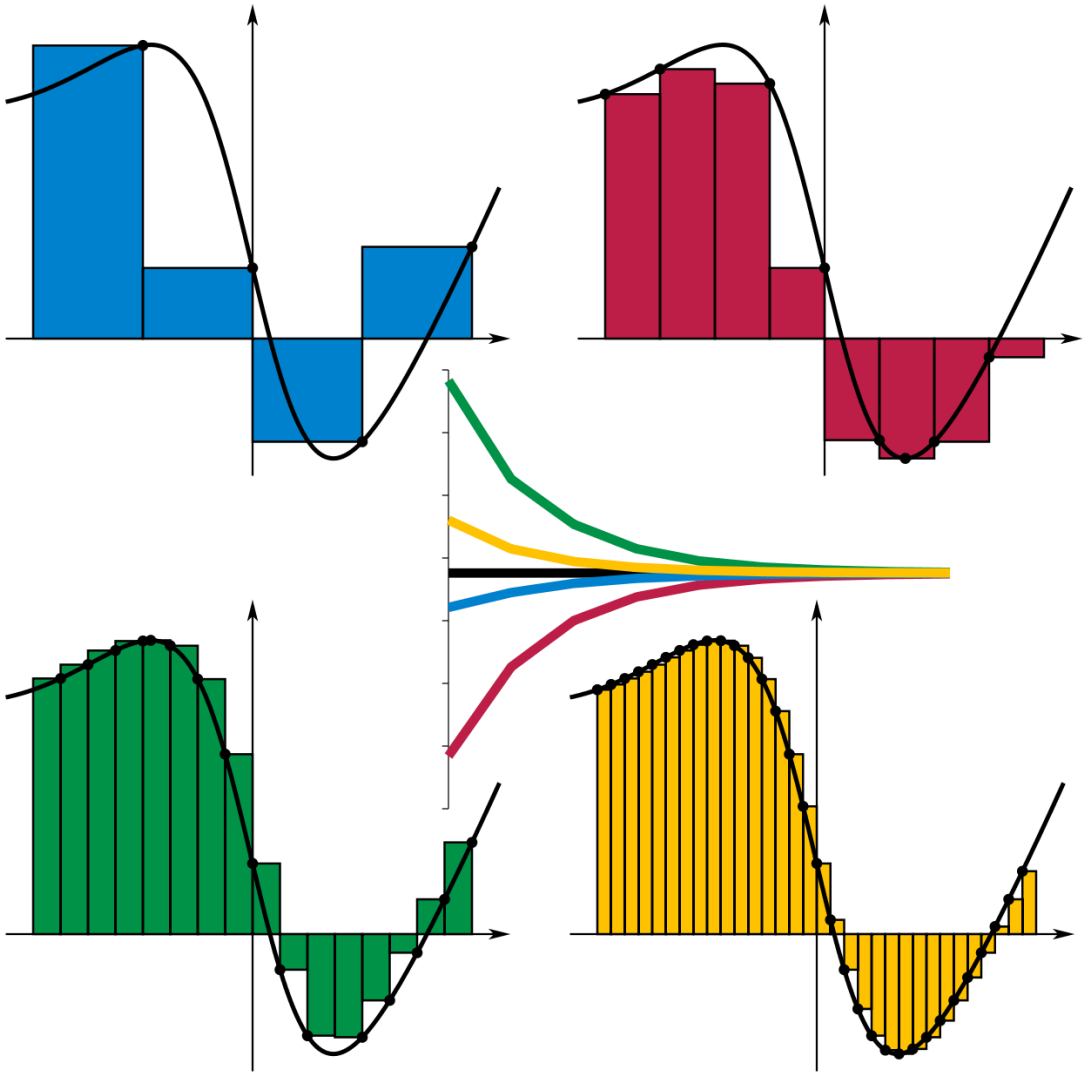
The system calls allowed are defined by a pointer to a Berkeley Packet Filter (BPF) passed via *args*. This argument is a pointer to a *struct sock_fprog*; it can be designed to filter arbitrary system calls and system call arguments. If the filter is invalid, **seccomp**() fails, returning **EINVAL** in *errno*.

If fork(2) or clone(2) is allowed by the filter, any child processes will be constrained to the same system call filters as the parent. If execve(2) is allowed, the existing filters will be preserved across a call to execve(2).

https://man7.org/linux/man-pages/man2/seccomp.2.html

Why normal samples

Microservices - Riemann sum



https://commons.wikimedia.org/wiki/File:Riemann_sum_convergence.png

How to find the fine-grained system calls?

LSMs

Might change the kernel. Not inherit
Profiles and how to write it.

- AppArmor
- SELinux
- TOMOYO Linux

LSMs

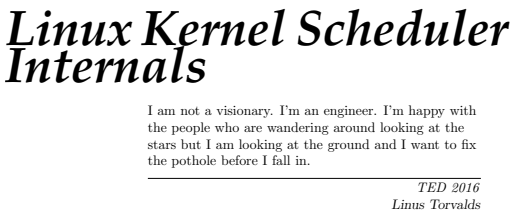
- AppArmor
  - Profiles
  - More fine-grained control in files.
- SELinux
  - Targeted, Role-Based-Access-Control, Multi-Layer/-Class Security
  - More fine-grained control in roles.
- TOMOYO Linux

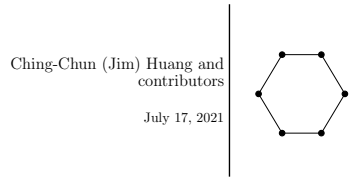
CGroups

It seems okay, but might be a little bit tricky to do it.

Reading, contributing.

Will be published in this weekend AT COSCUP.





CGroups

There are some people discuss when CGroups named CGroups from 'process containers'

1. 2008 cgroups: implement device whitelist cgrouplsm
2. 2011 cgroup: syscalls limiting subsystem
3. 2011 Limiting system calls via control groups?

   alonz: I wonder if it wouldn't be better to start from the other end of the solution space—small, incremental extensions to seccomp.

seccomp

- Advantage
  - Inherit
  - Block by each system call
- Disadvantage
  - Not cross platforms (CPU architecture)
  - Custom kernel

Papers

Intrusion Detection System for Applications Using Linux Containers [1]

- Toward Smart Moving Target Defense for Linux Container Resiliency [2]
- Smart Moving Target Defense for Linux Container Resiliency [3]
- Improving the Security of Microservice Systems by Detecting and Tolerating Intrusions [4]
- Malchain: Virtual Application Behaviour Profiling by Aggregated Microservice Data Exchange Graph [5]

BPF/eBPF

Tracee: Tracing Containers with eBPF
DockerCon19 - eBPF Superpowers
But they are not using in runtime container security.

Plans

Plans in next week

- Use the BPF feature and fuzzing technique to collect the "normal" system calls in healthcare data exchange system.
- Keep survey those papers.

References

[1] Amr S. Abed, Charles Clancy, and David S. Levy. "Intrusion Detection System for Applications Using Linux Containers". In: *Security and Trust Management*. Ed. by Sara Foresti. Cham: Springer International Publishing, 2015, pp. 123–135. ISBN: 978-3-319-24858-5.

[2] Mohamed Azab et al. "Toward Smart Moving Target Defense for Linux Container Resiliency". In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. 2016, pp. 619–622. DOI: 10.1109/LCN.2016.106.

[3] Mohamed Azab et al. "Smart Moving Target Defense for Linux Container Resiliency". In: *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. 2016, pp. 122–130. DOI: 10.1109/CIC.2016.028.

[4] José Flora. "Improving the Security of Microservice Systems by Detecting and Tolerating Intrusions". In: *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 2020, pp. 131–134. DOI: 10.1109/ISSREW51248.2020.00051.

[5] Mohammad Mahdi Ghorbani et al. "Malchain: Virtual Application Behaviour Profiling by Aggregated Microservice Data Exchange Graph". In: *2020 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. 2020, pp. 41–48. DOI: 10.1109/CloudCom49646.2020.00004.