健康資訊交換系統中之容器安全
The Container Security in Healthcare Data Exchange System

國立中山大學資訊工程學系

Department of Computer Science and Engineering

National Sun-Yet-San University, Taiwan

110 學年度大學部專題製作競賽

Bachelor's degree graduation project in 2021

Author: Chih-Hsuan Yang (B073040047)

Advisor: Chun-I Fan

October 10, 2021

Abstract

This research proposes a mechanism to enforce the system call a specific policy in the container, which is deployed in runtime. This policy is designed for the FHIR healthcare data exchange standard's container, which could guarantee the FHIR server does not have unsupported behavior and takes almost zero overhead. Recently, many companies use containers to run their microservices since containers could make their hardware resources be used efficiently. And the newest healthcare data exchange standard FHIR (Fast Healthcare Interoperability Resources) [1] has been implemented in a container by IBM, Microsoft, and Firebase. The deployment of FHIR in a container is a trend in the digital world [2] . However, containers are not sandboxes [3] . Containers are just isolated processes. Therefore, if hackers or malicious software could sneak into the container that would be a new cyber attacking surface in nearly future.

Contents

Chapter 1

Introduction

1.1 Container and Linux Kernel

The container is a secondary product of the operating system in the past 20 years. The FreeBSD develops ‘Jails’ in 1999, and the Solaris develops ‘Zones’ in 2004. Linux also took this idea into the Linux kernel, which is named cgroups (2007), the capabilities (2003), and seccomp (2005). However, why the Linux breaks this technology into many parts? This is because they had discussed: ”Why Should a System Administrator Upgrade?” in 2001 [4]. The Linux kernel almost entered the development path of ”upgrade for demand” like Microsoft Windows, and deviated from the original path of ”providing a mechanism but not a strategy” of the original Linux kernel.

While Linux were spreading in various server or embedded system devices, the Linux community got more pull requests to solved the scalability and virtualization issues [5]. However, they avoided confusion caused by multiple meanings of the term ”container” in the Linux kernel context. In kernel version 2.6.24 (2008) [6], control groups functionality was merged into the mainline, which is designed for an administrator (or administrative daemon) to organize processes into hierarchies of containers; each hierarchy is managed by a subsystem. Moreover, the cgroups was rewrote into cgroups-v2 in Linux kernel 4.5 (2015) [7].

The first and most complete implementation of the Linux container manager was LXC (Linux Containers). It was implemented in 2008 using cgroups and namespaces, and it runs on a single Linux kernel without requiring any patches. LXC provides a new view and imagination of virtualized services without any hypervisor. In 2016, Docker replaced LXC with ”libcontainer”, which was written in the Go programming language. Docker combined features in a new, more attractive way and made Linux containers popular.

The secondary product of the operating system, containers, offering many advantages: they enable you to ”build once, run anywhere.” Docker does this by bundling applications with all their dependencies into one package and isolating applications from the rest of the machine on which they’re running. Therefore, this research is based on docker container to propose a scheme of healthcare data exchange system’s security.

1.2 FHIR

FHIR is a standard for healthcare data exchange. The FHIR standard will be used in Taiwan in the near future. FHIR will be used to provide PHR (Personal Healthcare Records) in Taiwan. Therefore, we choose the most popular standard ”FHIR” for the target of the healthcare data exchange system.

1.2.1 RESTful API and Data Structure

REST (Representational State Transfer) is a stateless reliable web API, which is based on HTTP methods to access resources or data via URL parameters and the use of JSON or XML format to transmit queries. Because the RESTful is stateless, the client should keep their information (i.e. cookies) by themself.

FHIR has features: RESTful and data structure, make our research and benchmarks more accurate and reliable. Statelessness is a developer-friendly feature, the developer and the tester would not to design a complex state machine on the server-side or generating test files. And the FHIR takes RESTful as standard. Moreover, FHIR standard declared the ‘StructureDefinition’ [8]. These structure definitions are used to describe both the content defined in the FHIR specification itself - Resources, data types, the underlying infrastructural types, and also are used to describe how these structures are used in implementations.

1.2.2 Why IBM FHIR server

There are many applications using IBM’s FHIR server as the base component of the EHR (Electronic Health Records) system to communicate with the other various databases. Take it for example that the NextCloud’s EHR service, Taipei Veterans General Hospital, and AWS Cloud are using the FHIR server in a container for subroutine service.

NextCloud is an open-source and self-hosted productivity platform for users. Many people caring about their privacy issues distrust the FAANG (Facebook, Amazon, Apple, Netflix, Google), so they are using NextCloud to keep their privacy on their own. Therefore, they are eager to have a secure EHR system for their PHR (Personal Healthcare Records).

1.3 Data and Privacy

TBD.

Chapter 2

Related Works

2.1 Collecting System Calls

2.2 Find-granted Permission Control

2.3 Recently Exploited Vulnerabilities

2.3.1 Five Stage of Malware

2.4 Virtual Environment Performance Benchmark

Chapter 3

Preliminary

3.1 Container's Components

3.1.1 Namespaces

3.1.2 Cgroups

3.1.3 Seccomp

3.2 Programs in Execution

3.2.1 The task_struct in Kernel

3.2.2 Capabilities

3.3 User Mode Linux

3.3.1 Sandbox Security

3.3.2 gVisor

3.4 The (e)BPF

Chapter 4

Proposed Scheme

4.1 Workflow

4.1.1 Scan Base Image

4.1.2 Building and Signing

4.1.3 Check Image and Policy

4.1.4 Enforce the Policy

4.2 Rolling Updates

Chapter 5

Analysis and Benchmark

5.1 Analysis

5.1.1 Attacking Surface

5.1.2 Time Consuming

5.1.3 Statistics

5.2 Benchmark

5.2.1 Latency

5.2.2 Throughput

Chapter 6

Conclusion

6.1 Better Architecture

6.2 Future Machine Learning in Kernel

Reference

[1] HL7. *FHIR homepage*. URL: https://www.hl7.org/fhir/.

[2] A. Ahmed and G. Pierre. "Docker Container Deployment in Fog Computing Infrastructures". In: *2018 IEEE International Conference on Edge Computing (EDGE)*. 2018, pp. 1–8. DOI: 10.1109/EDGE.2018.00008.

[3] Ian Goldberg et al. "A Secure Environment for Untrusted Helper Applications Confining the Wily Hacker". In: *Proceedings of the 6th Conference on USENIX Security Symposium, Focusing on Applications of Cryptography - Volume 6*. SSYM'96. San Jose, California: USENIX Association, 1996, p. 1.

[4] informIT. *Version 2.4 of the LINUX KERNEL–Why Should a System Administrator Upgrade?* URL: https://www.informit.com/articles/article.aspx?p=20667.

[5] Silas Boyd-Wickizer et al. "An Analysis of Linux Scalability to Many Cores". In: *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*. Vancouver, BC: USENIX Association, Oct. 2010. URL: https://www.usenix.org/conference/osdi10/analysis-linux-scalability-many-cores.

[6] Jonathan Corbet. *Notes from a container*. URL: https://lwn.net/Articles/256389/.

[7] Tejun Heo. *Control Group v2*. URL: https://www.kernel.org/doc/Documentation/cgroup-v2.txt.

[8] HL7 FHIR. *Resource StructureDefinition*. URL: http://www.hl7.org/fhir/structuredefinition.html.