black hat EUROPE 2018

Cloud-Native Sandboxes for Microservices: Understanding New Threats and Attacks

Tongbo Luo (Chief AI Security Scientist, JD.com)

Zhaoyan Xu (Principal Security Researcher, Palo Alto Networks)

About Us

2014: Listed on Nasdaq (NASDAQ: JD)
2017: Fortune 500 Company (Rank 261)
2017: More than 1 Billion Active Customer
2018: Google Invest $500M

17K+
Merchant

16K+
Employee

Tongbo Luo - Chief AI Security Scientist - JD.com

About Us

Zhaoyan Xu (Principal Security Researcher, Palo Alto Networks)

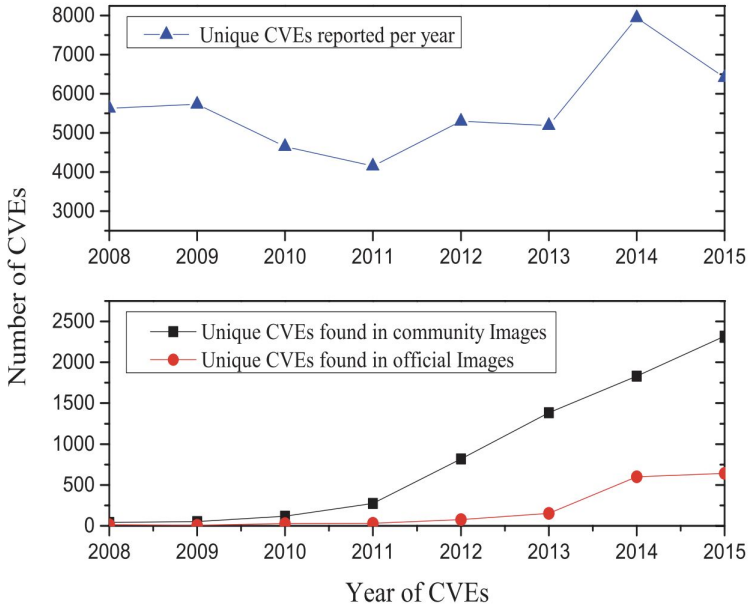
Agenda

- Introduction
- New Challenges
- System Design
  - Overview
  - Core Design Issue
- Parallel Execution and Alignment Analysis
- Case Study and Usage
  - Path Transversal
  - RCE
  - Authentication Bypass
  - Sandbox Escape

Introduction

- Container = Namespace + cGroup
- Cloud-native and container-based cluster
- Orchestrators - Kubernetes
- Container Security
  - Image Vulnerability
  - Docker/K8s Vulnerability
- Defense
  - Static Image Scanner
  - Dynamic Runtime Prevention/Detection

Image Type	Total Images	Number of Vulnerabilities				
		Mean	Median	Max	Min	Std. Dev.
Community	352,416	199	158	1,779	0	139
Community `:latest`	75,533	196	153	1,779	0	141
Official	3,802	185	127	791	0	145
Official `:latest`	93	76	76	392	0	59



Source: A Study of Security Vulnerabilities on Docker Hub

New Challenges

Problem-Side:

- Missing a novel sandbox tool to discover container-based threats.

Opportunity-side:

- Rethink the design of traditional sandbox.
- New Features to Improve the Detection.
  - Context-Awareness
  - Parallel Execution

Container Sandbox Design - Core Question

Q1: How to make it convenience to use sandbox in container-based cloud?

Q2: How to efficiently retrieve and build the context for sandbox?

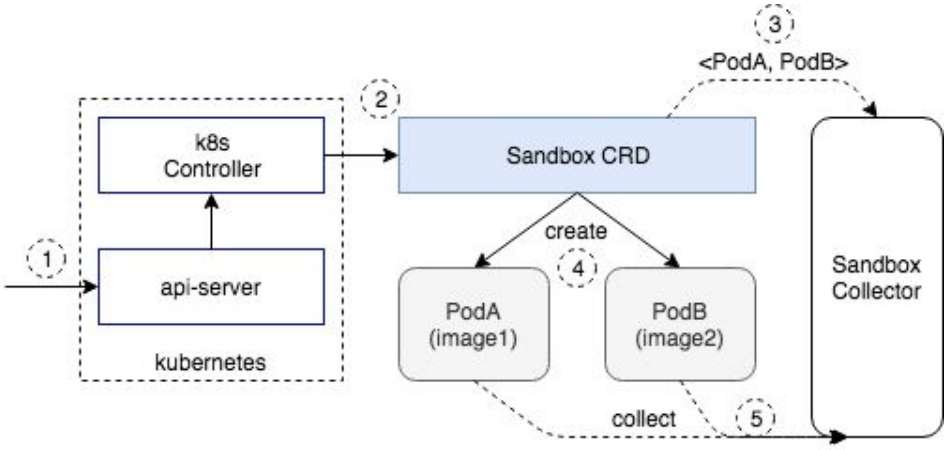
Q3: How to collect sample behaviors in sandbox?

Q4: How to analyze the sample behaviors beyond existing detection mechanism?

Q1 - Integrated into Orchestrator

- Using CRD to extend the k8s APIs
  - Introduce our custom-defined-resource type “sandbox”
  - No learning curve to manage container sandboxes.
- Example
  - Using YAML file to create a sandbox
  - With sample data (URL) and context info (image name)

```yaml
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: sandbox.contain.er
spec:
  group: contain.er
  version: v1alpha1
  names:
    kind: ParallelSandbox
    image1: tianon/exim4:latest
    image2: tianon/exim4:1.0
    sample:
      type: request
      URI: http://localhost:8080/vul?exploit
    singular: parallelsandbox
    plural: parallelsandboxes
```



Container Sandbox Design - Core Question

Q1: How to make it convenience to use sandbox in container-based cloud?

Q2: How to efficiently retrieve and build the context for sandbox?

Q3: How to collect sample behaviors in sandbox?

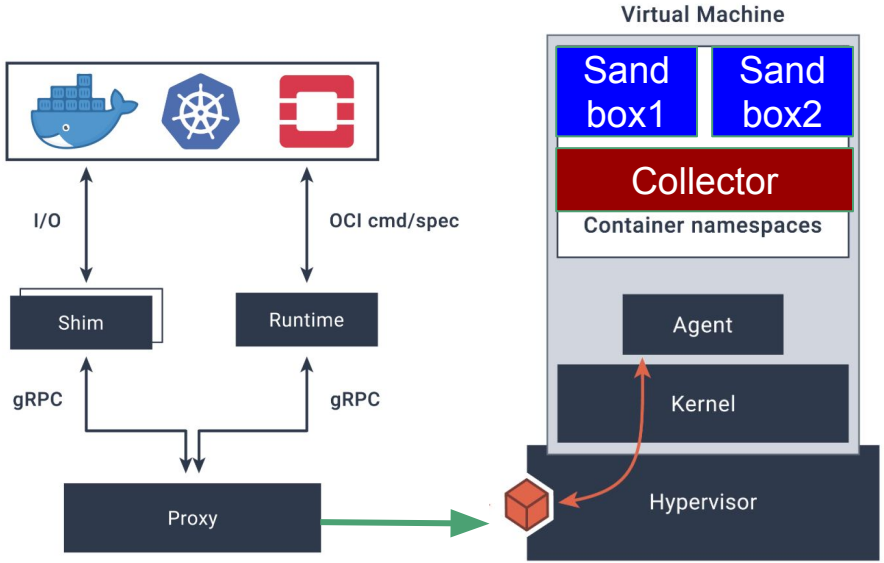
Q4: How to analyze the sample behaviors beyond existing detection mechanism?

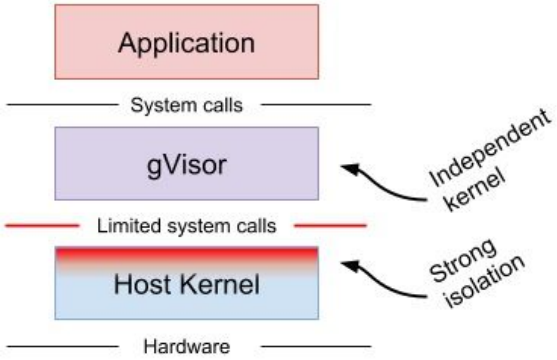
Q2 - EnvBuilder

- Rebuild the client's context for environment-sensitive detection.
- Input = Sample + Context
  - Explicit Way (User Upload DockerFile/Image Name)
  - Implicit Way (Retrieve Context from Orchestrators)

Q2 - Hard Isolation: Kata or gVisor

Run Sandbox in the Kata (Hypervisor-based Container Runtime)





Container Sandbox Design - Core Question

Q1: How to make it convenience to use in microservice cloud?

Q2: How to efficiently retrieve and build the context for sandbox?

Q3: How to collect sample behaviors in sandbox?

Q4: How to analyze the sample behaviors beyond existing detection mechanism?

Q3 - Syscall Collection

- **Sysdig**
  - based-on tracepoint + vring buffer
- eBPF -- enhancements to BPF (Berkeley Packet Filter)
  - BPF Compiler Collection (bcc)
  - Linux 4.x series
- auditd
  - Linux Audit system
  - Integrated to kernel since v2.6.9

Container Sandbox Design - Core Question

Q1: How to make it convenience to use in microservice cloud?

Q2: How to efficiently retrieve and build the context for sandbox?

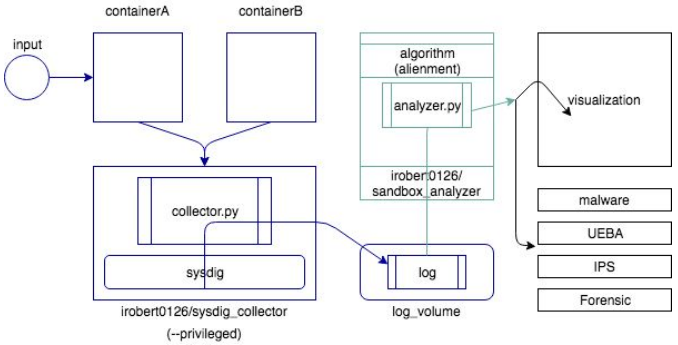
Q3: How to collect sample behaviors in sandbox?

Q4: How to analyze the sample behaviors beyond existing detection mechanism?

Parallel Execution and Alignment Analysis

Parallel Execution

1. Run Two containers created from the same image in parallel.
2. Feed two similar inputs to each container at the same time.
3. Collected behaviours.
4. Find the Differences between two behaviours.



Raw Data to Syscall Sequence

- **Raw Data**:

ruby3760A [27726, 21:43:14, >] [puma, 3818, 1, 14039, 18] [getsockopt, net] Data

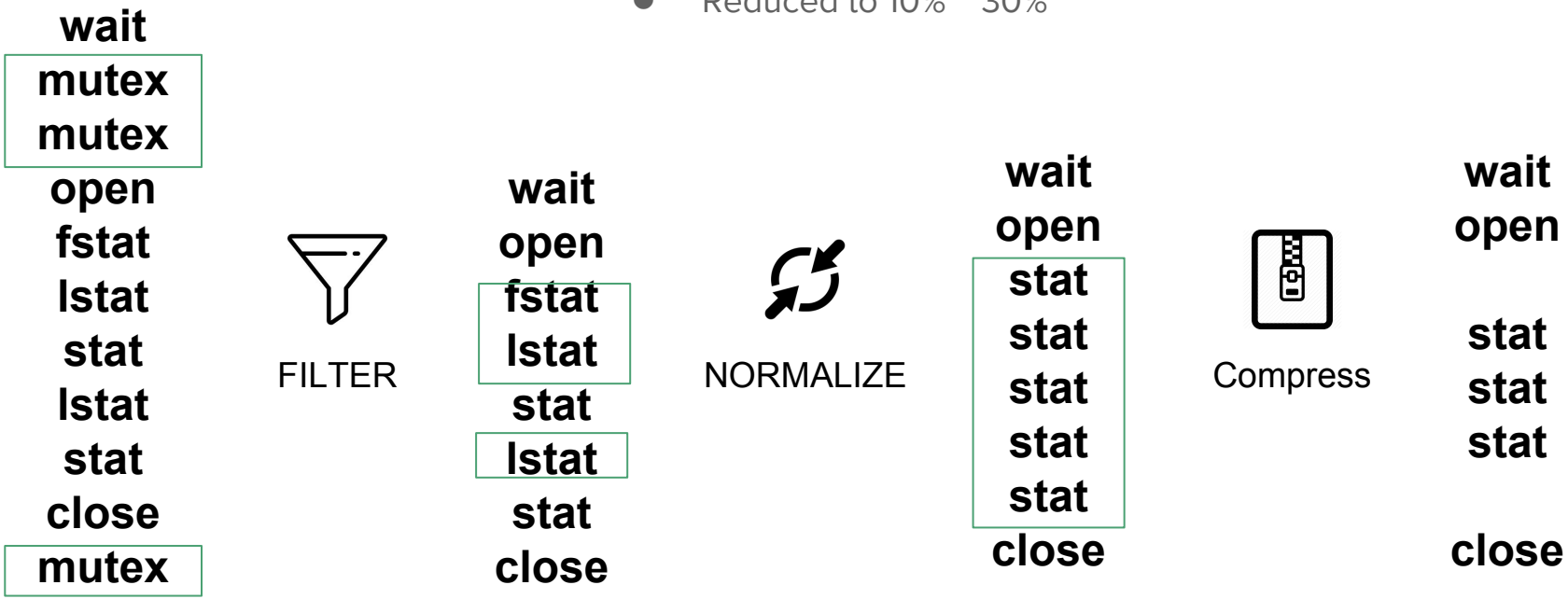
Container name	Metadata [event_num, time, direction]	Caller's Info [proc_name, pid, vpid, tid, vtid]	Syscall Info [call_type, call_catalog]	buffer

1. Group syscalls based on container-name and pid
2. Extract the syscall name
3. Map syscall name to an unique char

[open read write read read close] ➜ [1 a q a a 5]

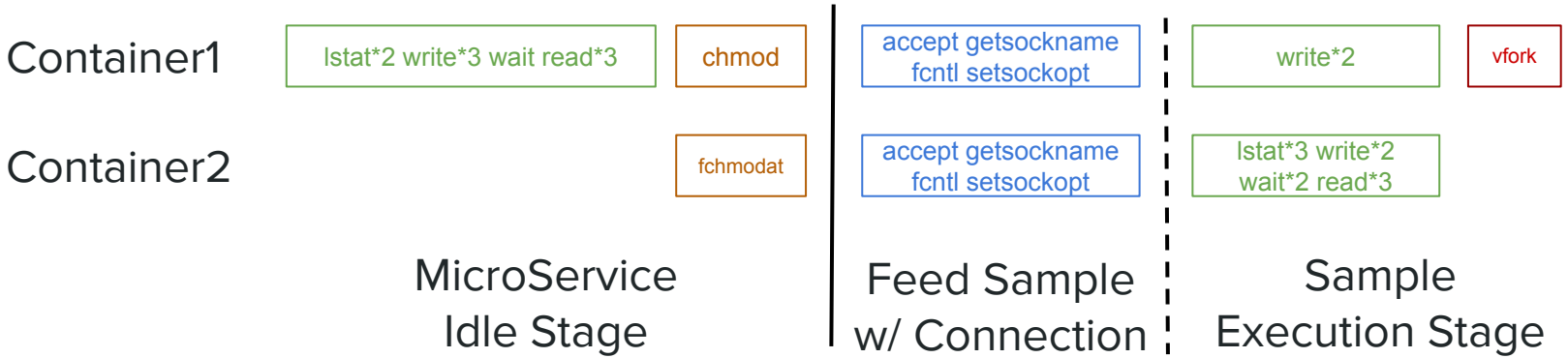
Performance Issue

- Syscall Sequence is too long
- 1000~5000 syscalls per second per process
- Reduced to 10% ~ 30%



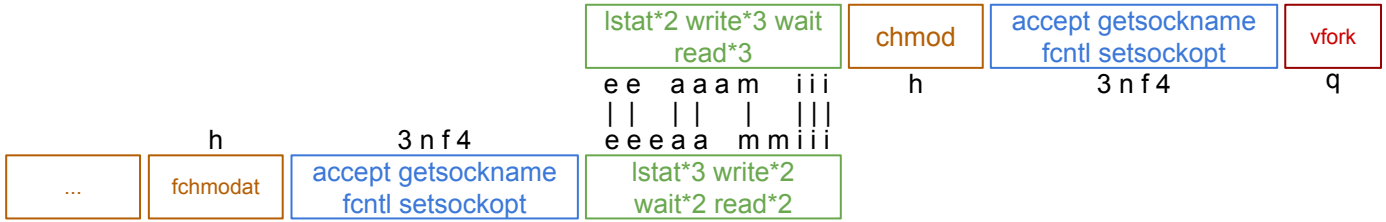
Scoring Function

- Quality of Scoring Function ➔ Quality of Alignment Result

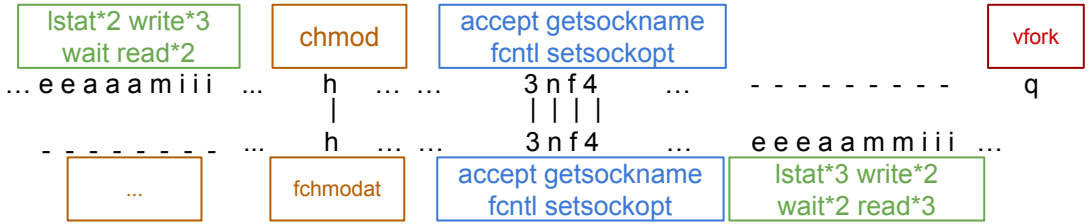


Scoring Function

Traditional Scoring Function Result



Ideal Result

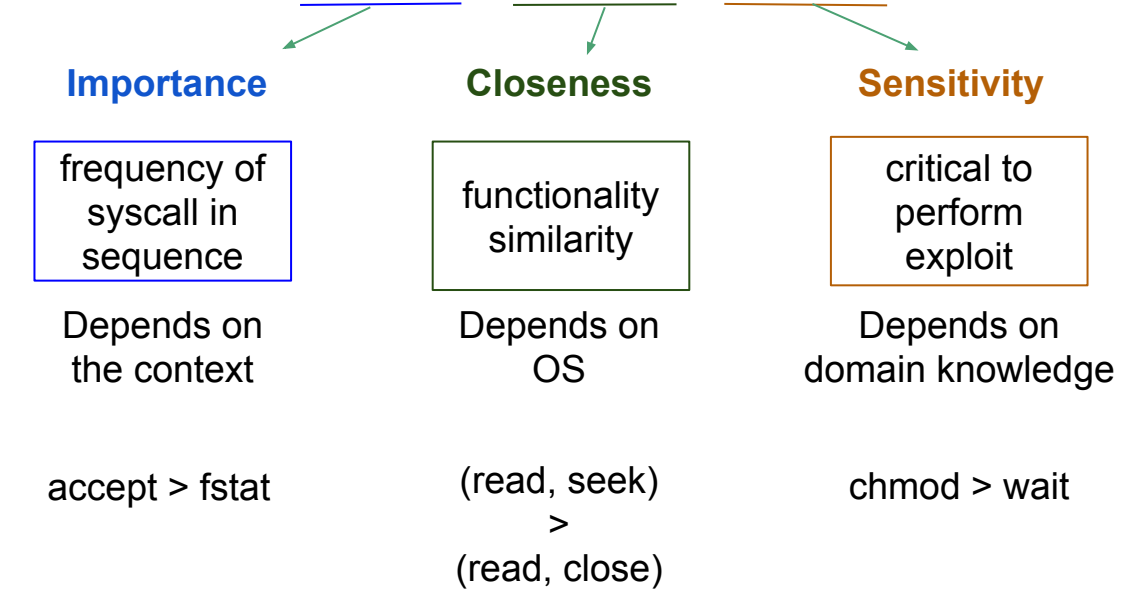


- Ideal Alignment Result: syscalls from the same stage are aligned.

Scoring Function

Our Customized Scoring Function for sandbox

$$score(sc1, sc2) = \Lambda(sc1, sc2) * \Phi(sc1, sc2) * \Psi(sc1, sc2)$$



Case Study and Practical Usage

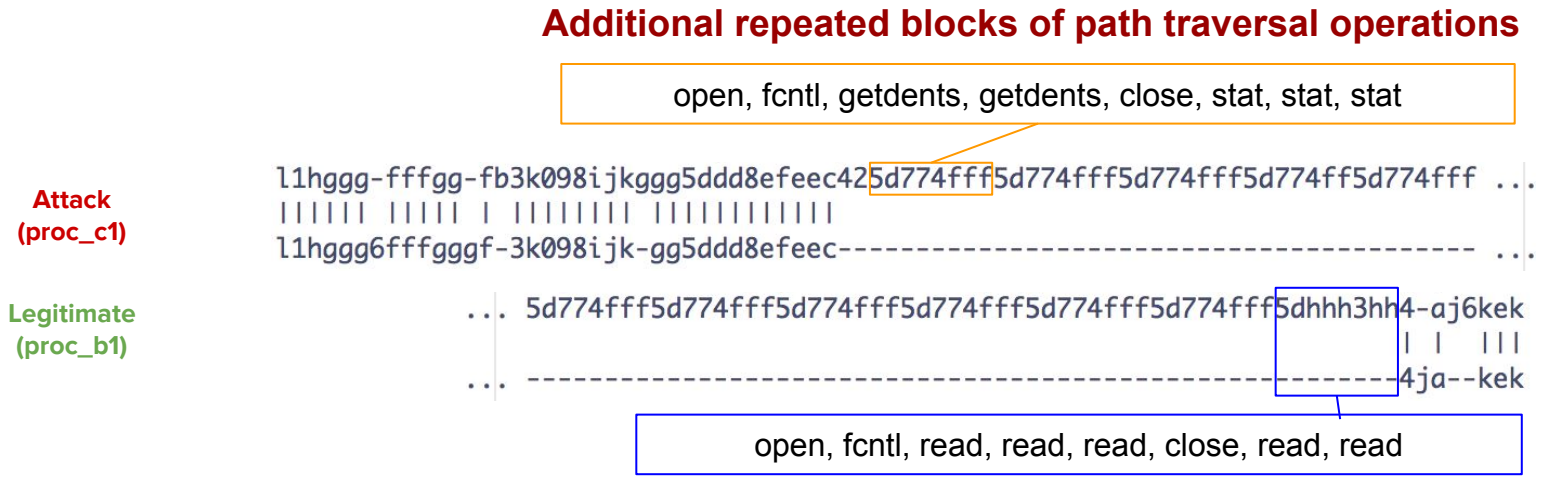
Effectiveness:

- Can we pinpoint exactly exploitation fraction from alignment result?
- Can we generalize a pattern for each type of attack?

Case Study - Path Traversal

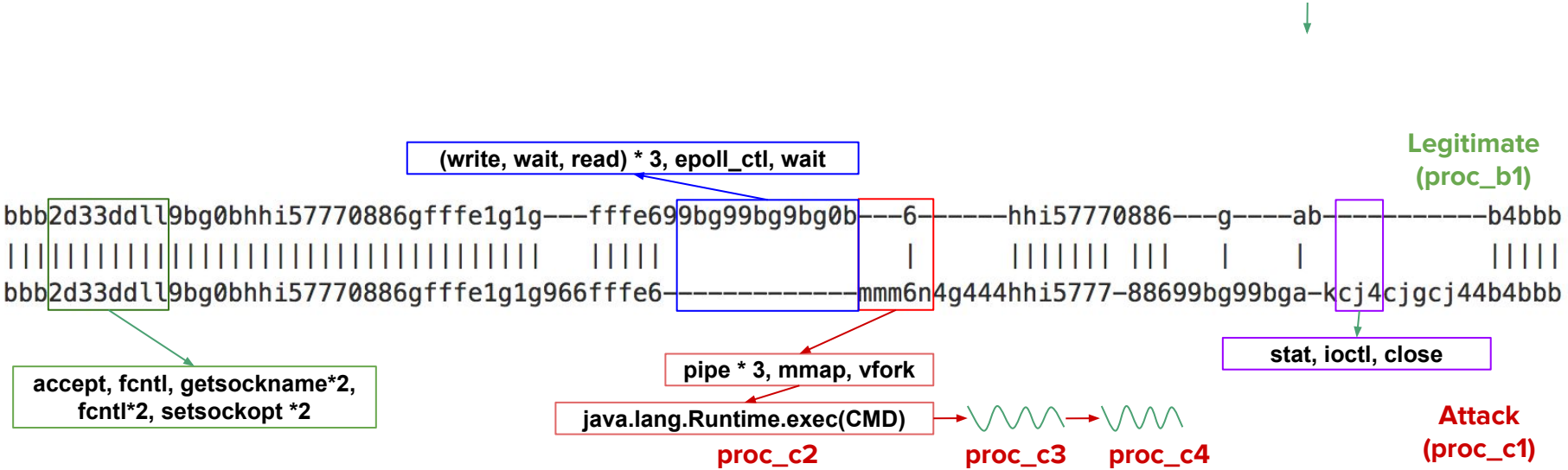
(CVE-2018-7490) @ uWSGI PHP plugin

attack `curl http://localhost:8080/..%2f..%2f..%2f..%2f..%2fetc/passwd`



Case Study - Remote Command Execution (RCE)

Vulnerability CVE-2016-4977 @ Spring Security OAuth



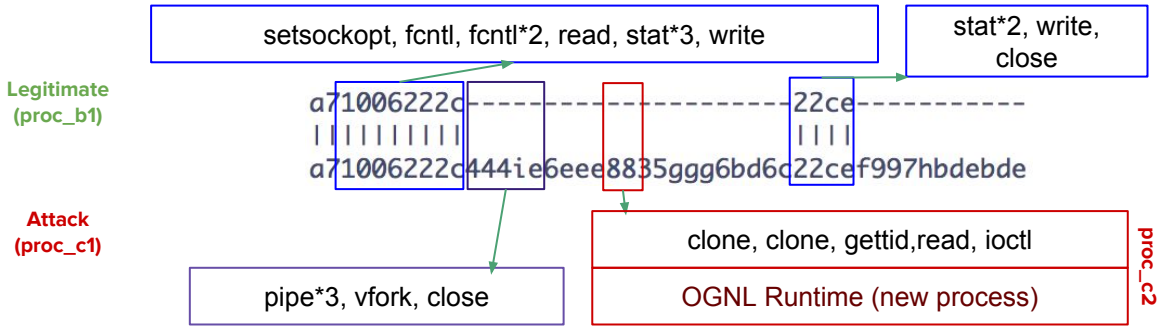
Case Study - Remote Command Execution (RCE)

Vulnerability CVE-2017-5638 @ Jakarta plugin in Apache Struts

- Content-Type arbitrary command execution

Embed injected OGNL script in “Content-Type” field from HTTP header

```
(#container=#context['com.opensymphony.xwork2.ActionContext.container'])
    (#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd}))
```



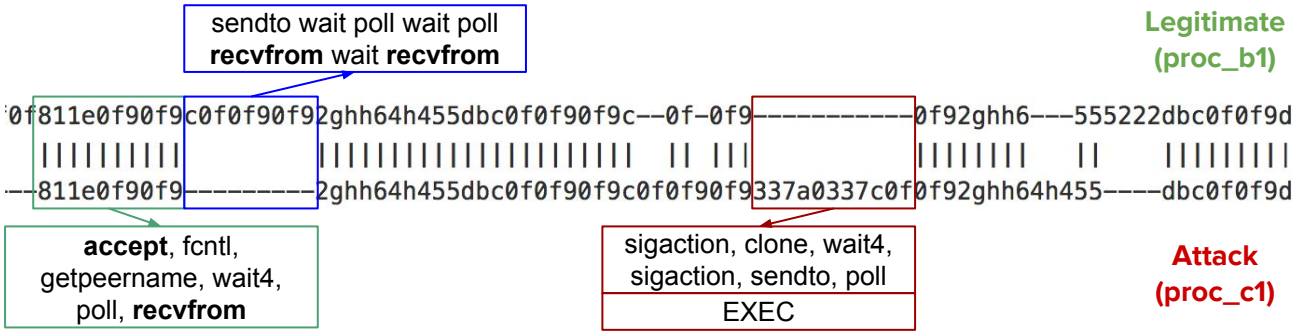
Case Study - Remote Command Execution (RCE)

Vulnerability CVE-2017-11610 @ Supervisord

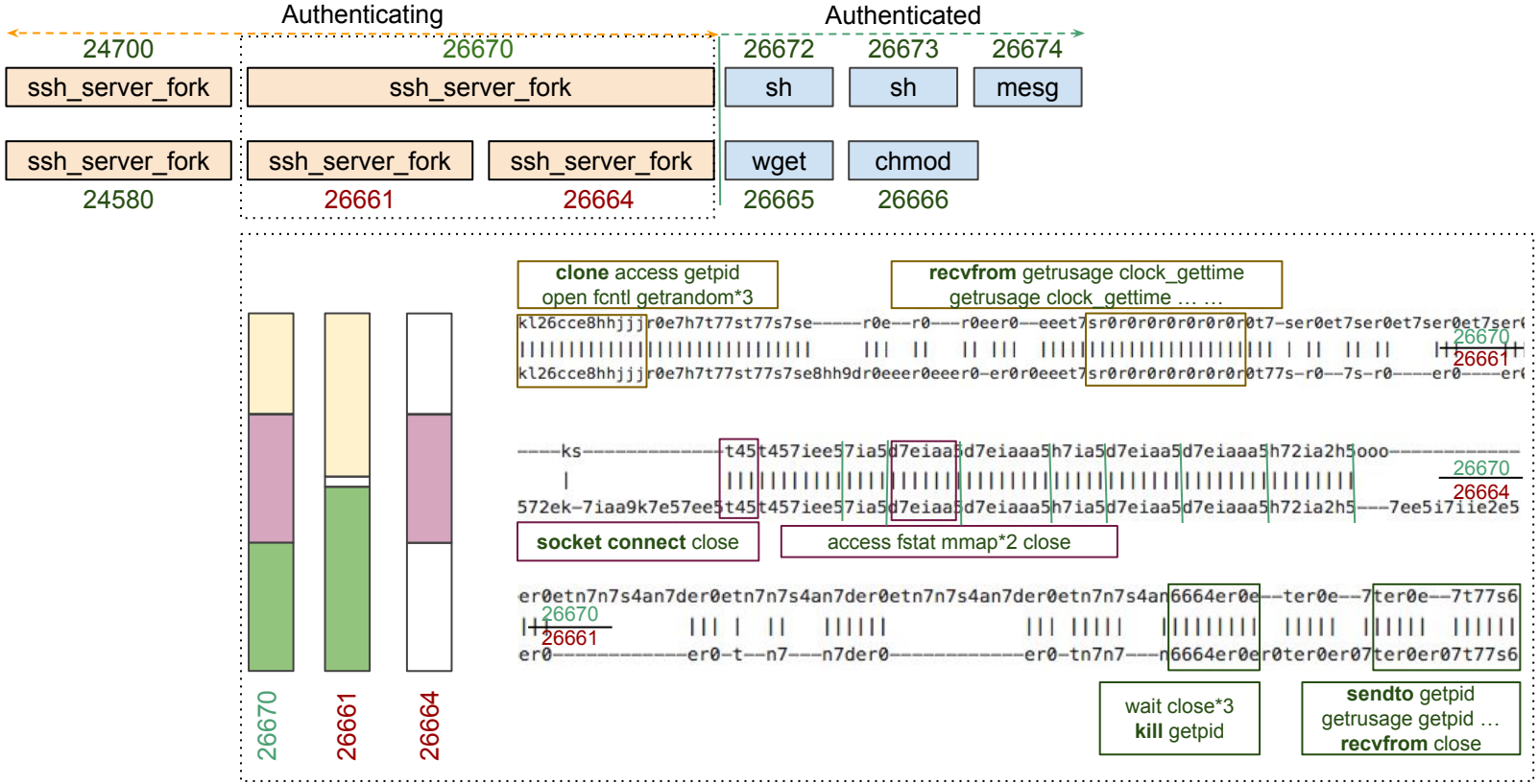
- client/server system to monitor/control processes on UNIX-like OSes.

container1: pids [12182 12120 6371 6372 6373]

container2: pids [12246 12306]



Case Study - Authentication Bypass

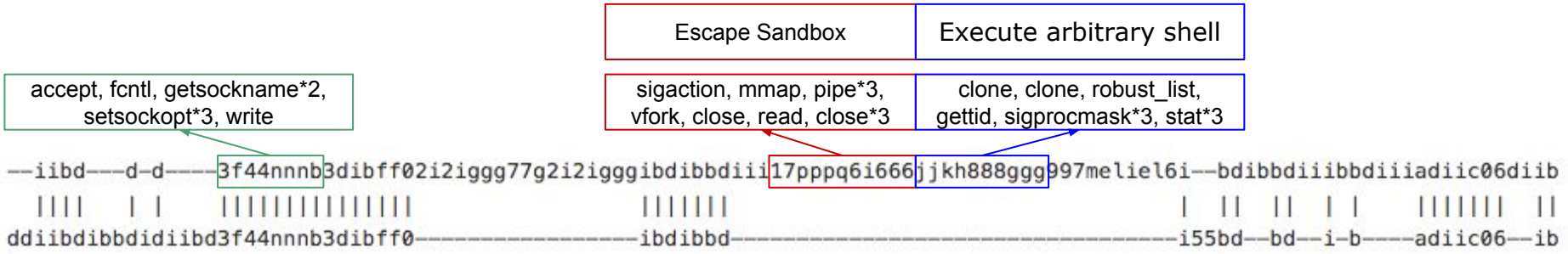


Case Study - JVM sandbox Escape

Vulnerability CVE-2015-1427 @ Elasticsearch

- full-text search engine
- Bypassing the Sandbox with Reflection:

java.lang.Math.class.forName("java.lang.Runtime").getRuntime().exec("id").getText()



- More detail in the white paper
  - Evaluation Result
  - Future work
- Q & A

paloalto NETWORKS®