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)

Agenda

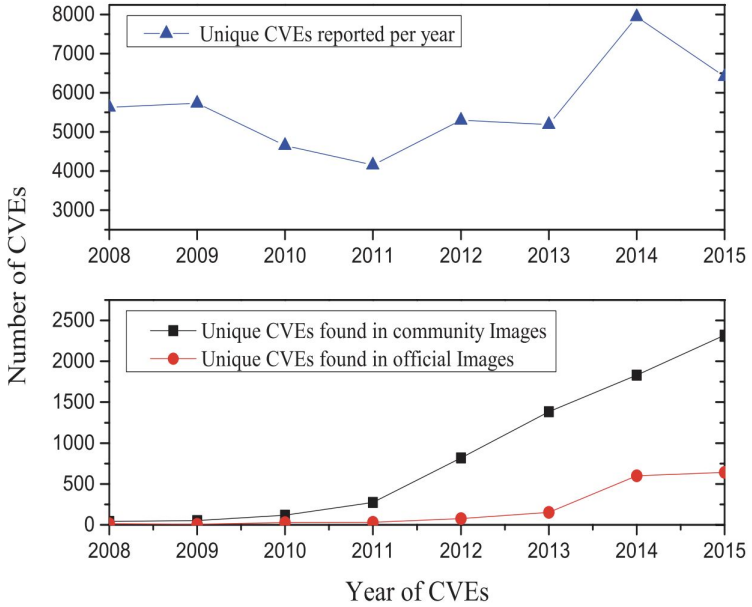
- Introduction
- Container Sandbox
- System Design
  - Overview
  - Integrated to K8s
  - Kata
- Parallel Execution and Alignment Analysis
- Case Study and Usage

Introduction

- Cloud-native and container-based cluster
- Orchestrators - Kubernetes
- Container Security
  - Image Vulnerability
  - Docker/K8s Vulnerability
- Defense
  - Static Image Scanning
  - Dynamic Runtime Prevention/Detection

Table 3: Number of Vulnerabilities per Image.

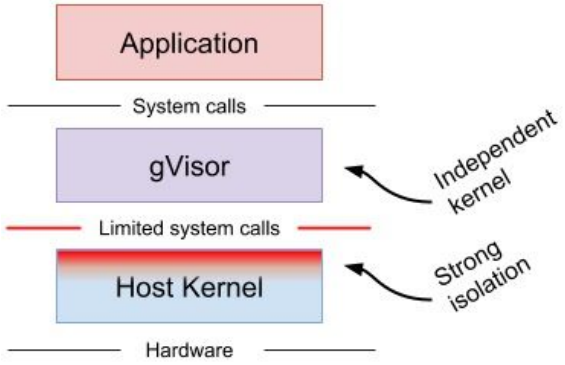
Image Type	Total Images	Number of Vulnerabilities				
		Mean	Median	Max	Min	Std. Dev.
Community	352,416	199	158	1,779	0	139
Community `:latest`	75,533	196	153	1,779	0	141
Official	3,802	185	127	791	0	145
Official `:latest`	93	76	76	392	0	59

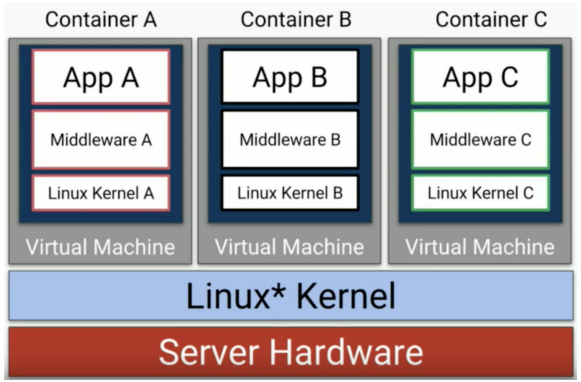


Source: A Study of Security Vulnerabilities on Docker Hub

Container Sandbox

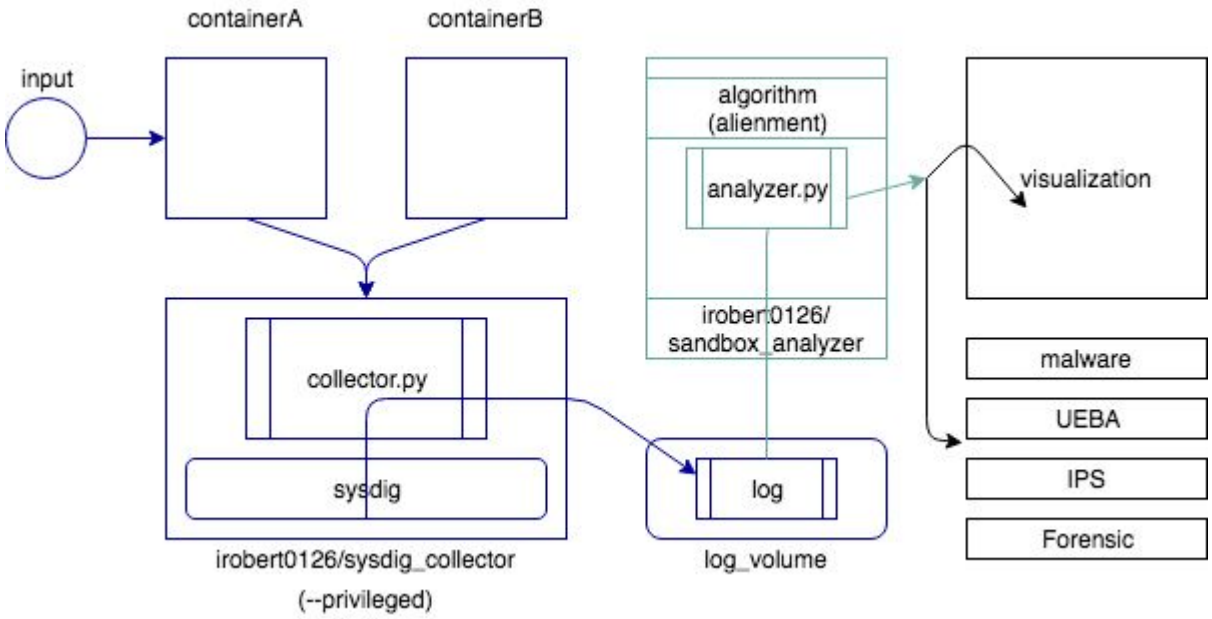
- Sandboxing is a proven technique for detecting malware and targeted attacks.
- Traditional VM sandbox 🆚 Container sandbox
- Hard Isolation: gVisor, Kata





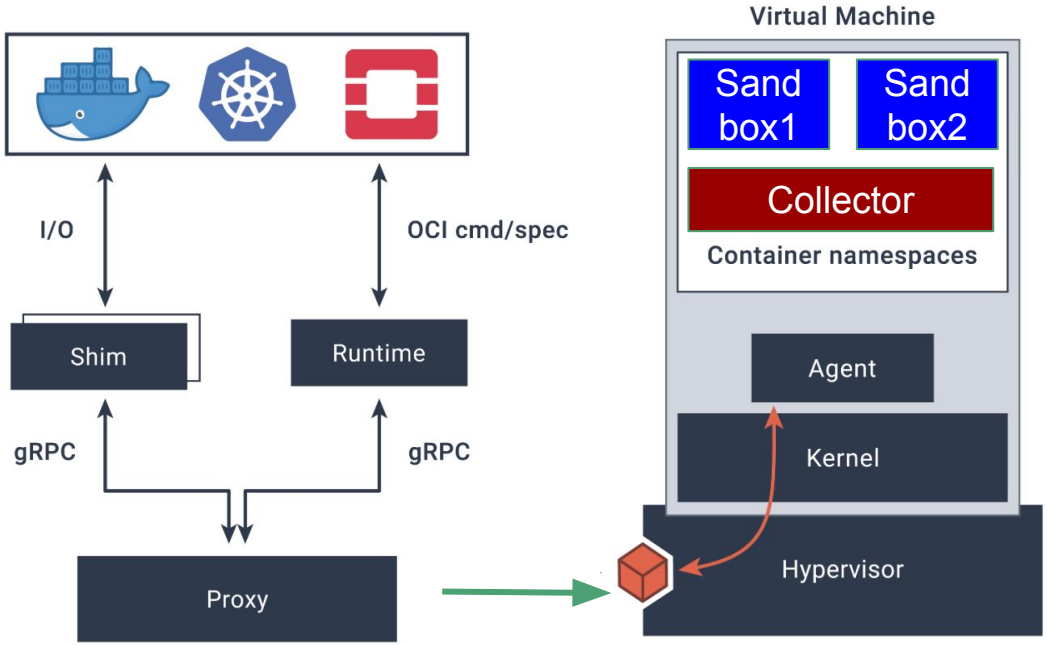
System Design

Overview



System Design - Enable Kata

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

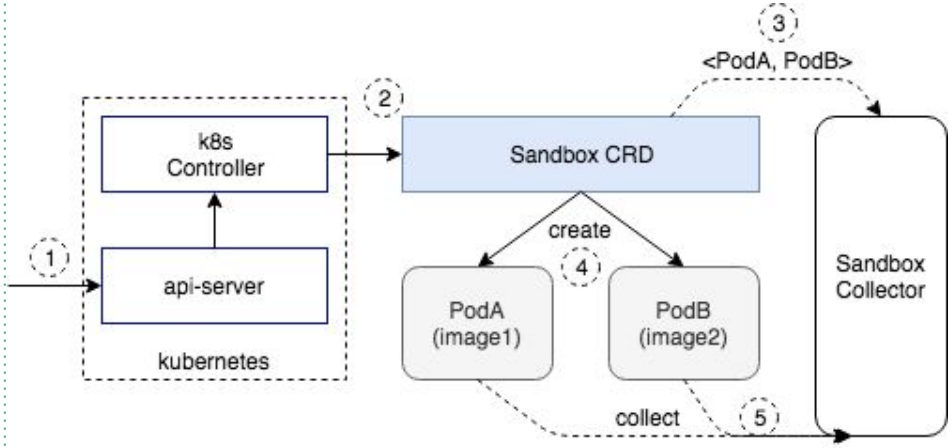


System Design - Integrated with k8s

- CRD
  - Custom Resource Definitions
  - A custom resource is an extension of the Kubernetes API
  - https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/
- Example

```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
```

- Deployment



System Design - 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

Parallel Execution and Alignment Analysis

Parallel Execution

Run Two containers created from the same image in parallel.

Feed two similar inputs to each container at the same time.

Collected behaviours among two containers.

Find the Differences between two behaviours.

Alignment Analysis 0

Collected System Calls To Syscall Sequence.

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

Covert Syscall names to single unique character

In order to leverage existing DNA sequence alignment tools

Mapping: { syscall : char }

Alignment Analysis 1

Problem: Syscall Sequence is so long, causing the alignment takes forever.

  - 1000~5000 syscalls per second per process
  - Dynamic Programming: O(m x n) + O(max(m, n))
- Filter:
  - Filter out: futex, mprotect
- Normalize:
  - Normalized similar calls: [stat, fstat, lstat] => stat
- Compress:
  - Compress the continually identical system calls from n to 3.
  - Example: [open, open, …], [stat, lstat, ...]
- Repeated Pattern:

Alignment Analysis 2

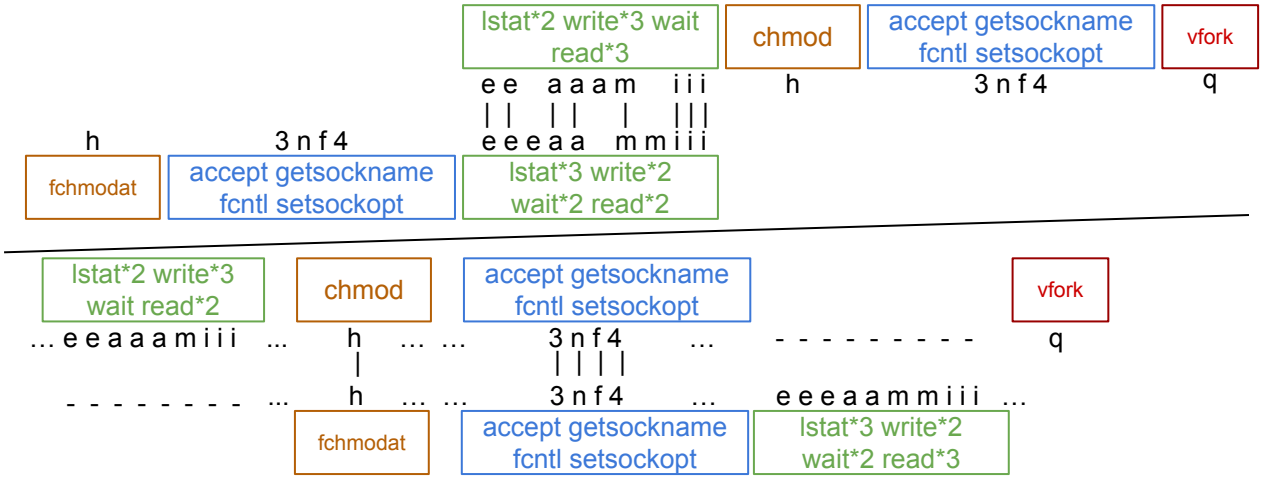
Problem: Precise

Scoring Function based on our context

How to define matching:

Matching Rewards:

Alignment with default scoring function



Alignment with customized scoring function

Case Study and Practical Usage

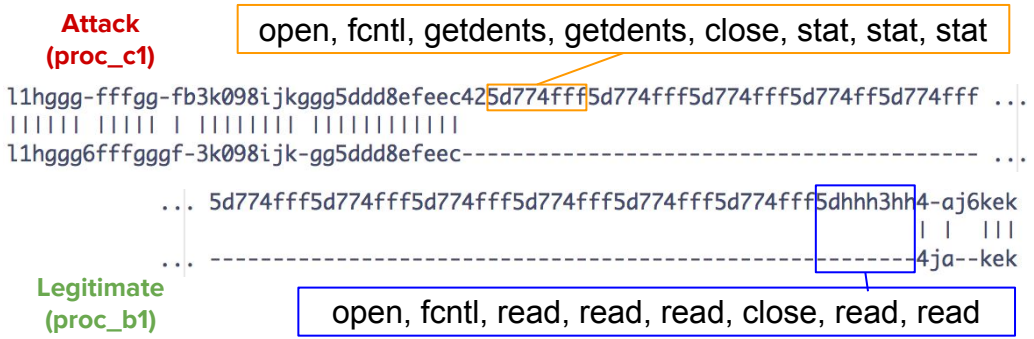
Case Study - CVE-2018-7490

uWSGI PHP - a web application

Vulnerability - Path Traversal Vulnerability

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

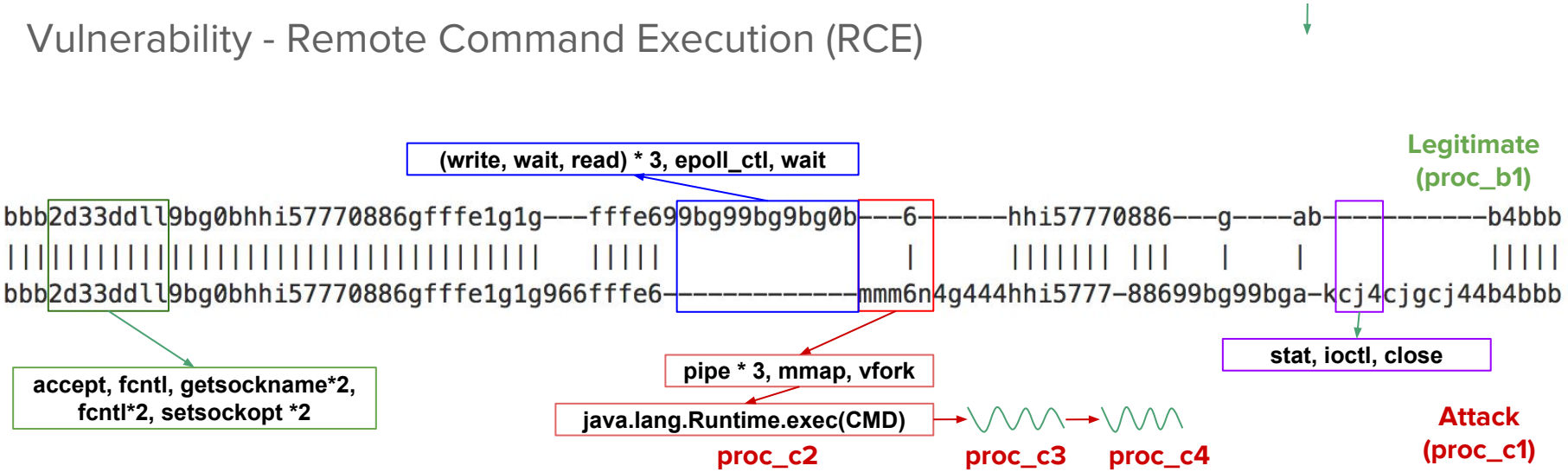
benign curl http://localhost:8080/



Case Study - CVE-2016-4977

Spring Security OAuth

Vulnerability - Remote Command Execution (RCE)



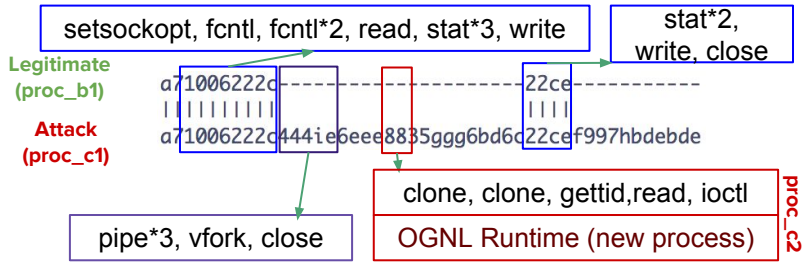
Case Study - CVE-2017-5638

Apache Struts - vulnerable Jakarta plugin

Vulnerability - 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}))
```



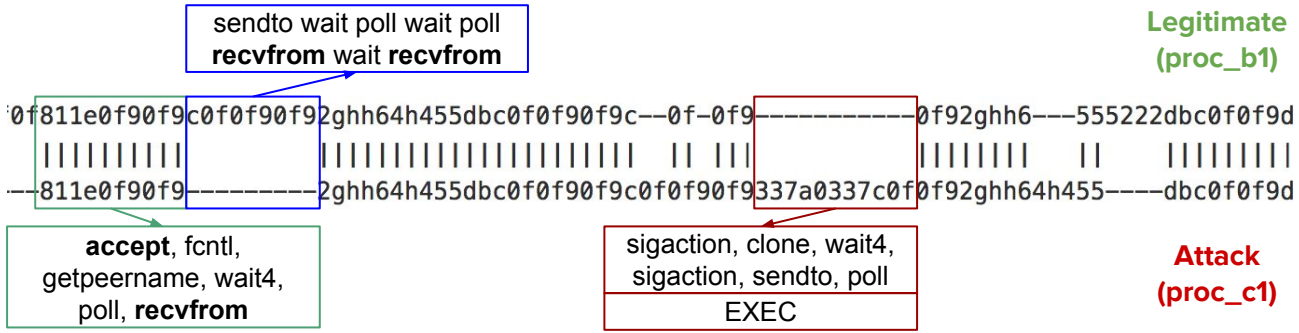
Case Study - CVE-2017-11610

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

Vulnerability - Remote Command Execution (RCE)

container1: pids [12182 12120 6371 6372 6373]

container2: pids [12246 12306]



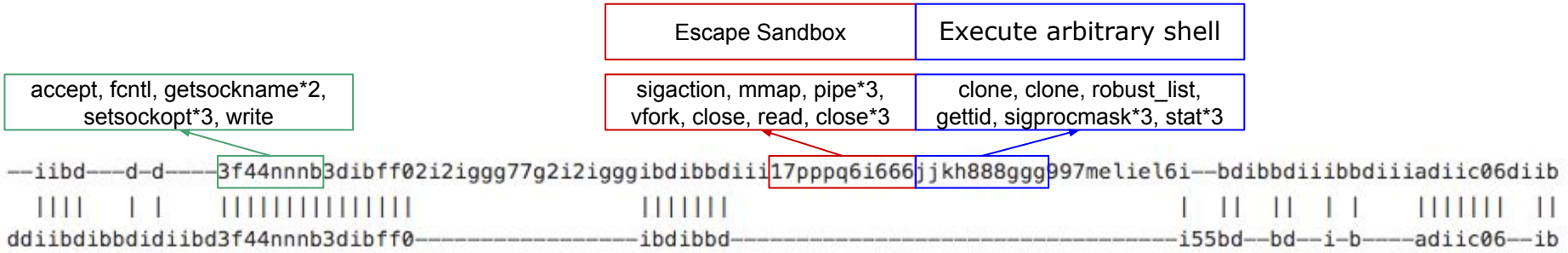
Case Study - CVE-2015-1427

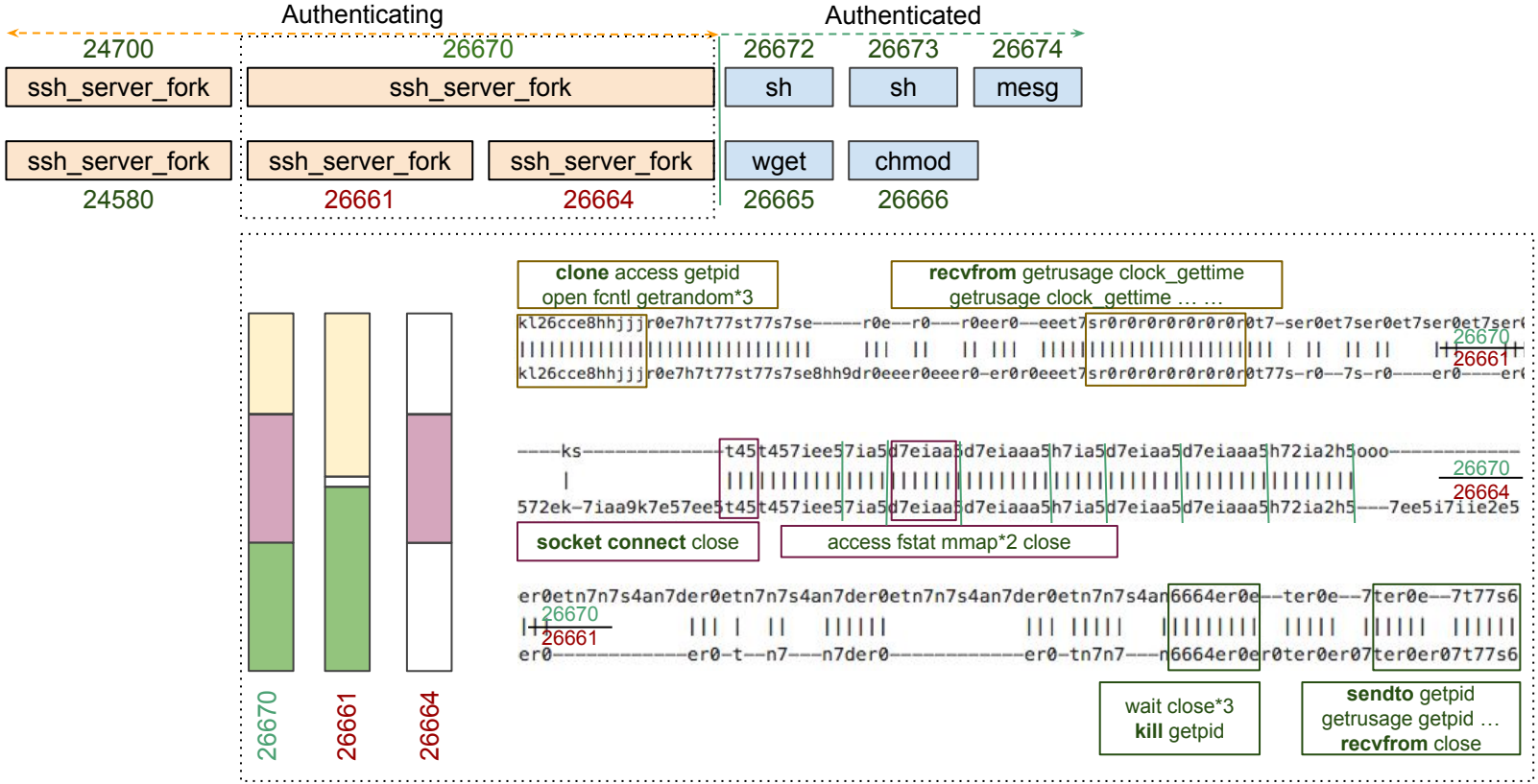
Elasticsearch - full-text search engine

Vulnerability - JVM sandbox Escape

Bypassing the Sandbox with Reflection:

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




Authenticating
Authenticated
24700
26670
26672
26673
26674
ssh_server_fork
ssh_server_fork
sh
sh
mesg
ssh_server_fork
ssh_server_fork
ssh_server_fork
wget
chmod
24580
26661
26664
26665
26666
clone access getpid
open fcntl getrandom*3
recvfrom getrusage clock_gettime
getrusage clock_gettime … …
socket connect close
access fstat mmap*2 close
wait close*3
kill getpid
sendto getpid
getrusage getpid …
recvfrom close
26670
26661
26664

Q & A