Getting a Good Night’s Sleep
After a Fresh Kubernetes Deployment

Paweł Wieczorek
August 21, 2019

Samsung R&D Institute Poland
Outline

Motivation

Resources

Utilities

Proof of Concept

Summary
Motivation
What is ONAP?

- Platform for network service orchestration and automation
- Targeted at telecommunication, cable and cloud service providers
- More on https://docs.onap.org
Static analysis and vulnerability assessment

https://sonar.onap.org
ONAP Penetration Test Report

https://wiki.onap.org/display/DW/ONAP+Casablanca+Security+Assessment
ONAP Security Advisories

- OSA-2019-026: AAF Secret Management Service allows to access all stored data
- OSA-2019-025: Unprotected APIs/UIs exposed in CLI project
- OSA-2019-024: Unprotected APIs/UIs exposed in MSB project
- OSA-2019-023: Unprotected APIs/UIs exposed in SO project
- OSA-2019-022: Unprotected APIs/UIs exposed in OOM project
- OSA-2019-021: Unprotected APIs/UIs exposed in DCAE project
- OSA-2019-020: Unprotected APIs/UIs exposed in Logging project
- OSA-2019-019: SDNC service allows for arbitrary code execution in sla/upload form
- OSA-2019-017: Some ONAP services allows to impersonate any user without authentication

https://git.onap.org/osa

https://docs.onap.org/en/latest/submodules/osa.git/docs/osalist.html
Current involvement

ONAP Architecture
Version 4.0.10
Date: May 16th, 2019

Design-Time
- VNF Validation
- VNF SDK
- VNP
- Service Design & Creation (SDC)
  - Service/xNF Design
  - xNF Onboarding
  - Workflow Designer
  - Controller Design Studio
  - DCAE Design Studio
- Catalog

Run-Time
- Portal
- O&M Dashboard (VID)
- Use-Case UI
- External APIs
- CLI
- Interfaces
- Control Loop Automation (CLAMP)
- Policy Framework
- Service Orchestration (SO)
- Active & Available Inventory (AAI)
- External System Register (ESR)
- Microservice Bus (MSB) / Message & Data Routers (DMaaS)

Manage ONAP
- ONAP Operations Manager (OOM)

ONAP Operations
- Dublin

Shared Services
- AuthN/AuthZ (AAF)
- Optimization (OOF)
- Logging
- Audit (POMBA)
- Multi-Site State (MUSIC)
- Others...

External Systems
- Third Party Controllers
- sVNF
- EMS

Network Function Layer
- VNFs
- PNs

Hypervisor / OS Layer
- OpenStack
- Commercial VIM
- Kubernetes
- Public Cloud

Managed Environment
- MPLS
- Private Cloud
- Private DC Cloud
- IP
- Public Cloud
Resources
Securing a Cluster

This document covers topics related to protecting a cluster from accidental or malicious access and provides recommendations on overall security.

- Before you begin
- Controlling access to the Kubernetes API
- Controlling access to the Kubelet
- Controlling the capabilities of a workload or user at runtime
- Protecting cluster components from compromise

https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster/
Security Committee and Security Audit Working Group

- https://github.com/kubernetes/security
- https://github.com/kubernetes/community/tree/master/wg-security-audit
Others’ experience

- Books (e.g. *Kubernetes Security* by Liz Rice and Michael Hausenblas)
- Guides (e.g. https://github.com/freach/kubernetes-security-best-practice)
- Blogs (e.g. https://sysdig.com/blog/33-kubernetes-security-tools)
- Benchmarks (https://www.cisecurity.org/cis-benchmarks)
Utilities
<table>
<thead>
<tr>
<th>Feature</th>
<th>Lite v3</th>
<th>Lite v4</th>
<th>Pro v3</th>
<th>Pro v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmarks</td>
<td>several</td>
<td>several</td>
<td>90+</td>
<td>65+</td>
</tr>
<tr>
<td>GUI</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>CLI</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
## Advantages

- Generic approach: Chef InSpec Profile

## Limitations

- Depends on external tool for execution

## Project

https://github.com/dev-sec/cis-kubernetes-benchmark
### Advantages

- Single binary
- Tests configured with YAML files

### Limitations

- Complex execution (unless cluster is modified)

### Project

https://github.com/aquasecurity/kube-bench
Proof of Concept
Requirements

- No changes on cluster
- External tools kept at minimum
- Supports ONAP Casablanca and Dublin
### Cluster configuration

- Casablanca: Rancher CLI
- Dublin: RKE cluster definition (*cluster.yml*)

### Node interaction

- Casablanca: Rancher CLI
- Dublin: SSH
Setup, deployment and execution

- Go: single binary
- Standard Go toolkit (with convenience `make` targets)
- All the data collected prior analysis

https://git.onap.org/integration/tree/test/security/k8s
Summary
Future work

- Configurable benchmarks
- Benchmark recommendations
- Cluster configuration improvement proposals
- Upstreaming default execution configurations
Conclusion

- Tailored to specific needs
- Might be beneficial to other projects
- Room for improvement and collaboration