



Соответствует ли ваш Kubernetes-кластер лучшим практикам?

Дмитрий Евдокимов
Founder&CTO Luntry

Обо мне



- Основатель и технический директор [Luntry](#)
- Опыт в ИБ более 10 лет
- Специализация безопасность контейнеров и Kubernetes
- CFP DevOpsConf, HighLoad++
- Бывший автор статей и редактор рубрик в журнале "ХАКЕР"
- Автор Telegram-канала "[k8s \(in\)security](#)"
- Автор курса "Cloud Native безопасность в Kubernetes"
- Не верит, что систему можно сделать надежной и безопасной, не понимая ее
- Организатор конференции по безопасности контейнеров - [БеКон](#)
- Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, OFFZONE, PHDays, Kazhackstan, DevOpsConf, DevOops, KuberConf, VK Kubernetes Conference, HighLoad++, БЕКОН и др.

О компании Luntry

- Luntry – это Комплексная Защита на всем жизненном цикле контейнерных приложений и средств оркестрации на базе Kubernetes
- Продукт в реестре Минцифры
 - <https://reestr.digital.gov.ru/reestr/1057835/>
- В процессе получения сертификата ФСТЭК
 - Ориентировочно второй квартал 2024



Функциональность Luntry



Функциональность Luntry



План вебинара

- Стандарты безопасности для контейнерных сред
- Очевидные и не очень вещи в CIS Kubernetes Benchmark
- Возможности Open Source решений
- Взгляд Luntry

Compliance

Cluster Name Gateway

Group by subsections 14.02.2024 13:00 cis:1.8 Select date range Export 0 Checks: 149 Statuses: 4 Severities: 4

62.04 %
61.78 %
61.63 %

13.02.24 14:00 13.02.24 16:00 13.02.24 18:00 13.02.24 20:00 13.02.24 22:00 14.02.24 00:00 14.02.24 02:00 14.02.24 04:00 14.02.24 06:00 14.02.24 08:00 14.02.24 10:00 14.02.24 13:00

5. Policies 35 Controls (3 Pass) (29 Fail) (0 Error) (3 Warn)

5.1. RBAC and Service Accounts 13 Controls (1 Pass) (11 Fail) (0 Error) (1 Warn)

ID	Status	Severity	Name	Success rate	Resources Count
> 5.1.1	FAIL	HIGH	Ensure that the cluster-admin role is only used where required	96.34 %	3
> 5.1.2	FAIL	MEDIUM	Minimize access to secrets	81.71 %	15
> 5.1.3	FAIL	HIGH	Minimize wildcard use in Roles and ClusterRoles	57.32 %	35
> 5.1.4	FAIL	MEDIUM	Minimize access to create pods	89.02 %	9
> 5.1.5	PASS	MEDIUM	Ensure that default service accounts are not actively used	100.00 %	0
> 5.1.6	FAIL	MEDIUM	Ensure that Service Account Tokens are only mounted where necessary	1.69 %	267

Profile applicability: Level 1 - Master Node

Description: Service account tokens should not be mounted in pods except where the workload running in the pod explicitly needs to communicate with the API server

Rationale: Mounting service account tokens inside pods can provide an avenue for privilege escalation attacks where an attacker is able to compromise a single pod in the cluster. Avoiding mounting these tokens removes this attack avenue.

Impact: Pods mounted without service account tokens will not be able to communicate with the API server, except where the resource is available to unauthenticated principals.

Remediation: Modify the definition of pods and service accounts which do not need to mount service account tokens to disable it.

Стандарты безопасности для контейнерных сред



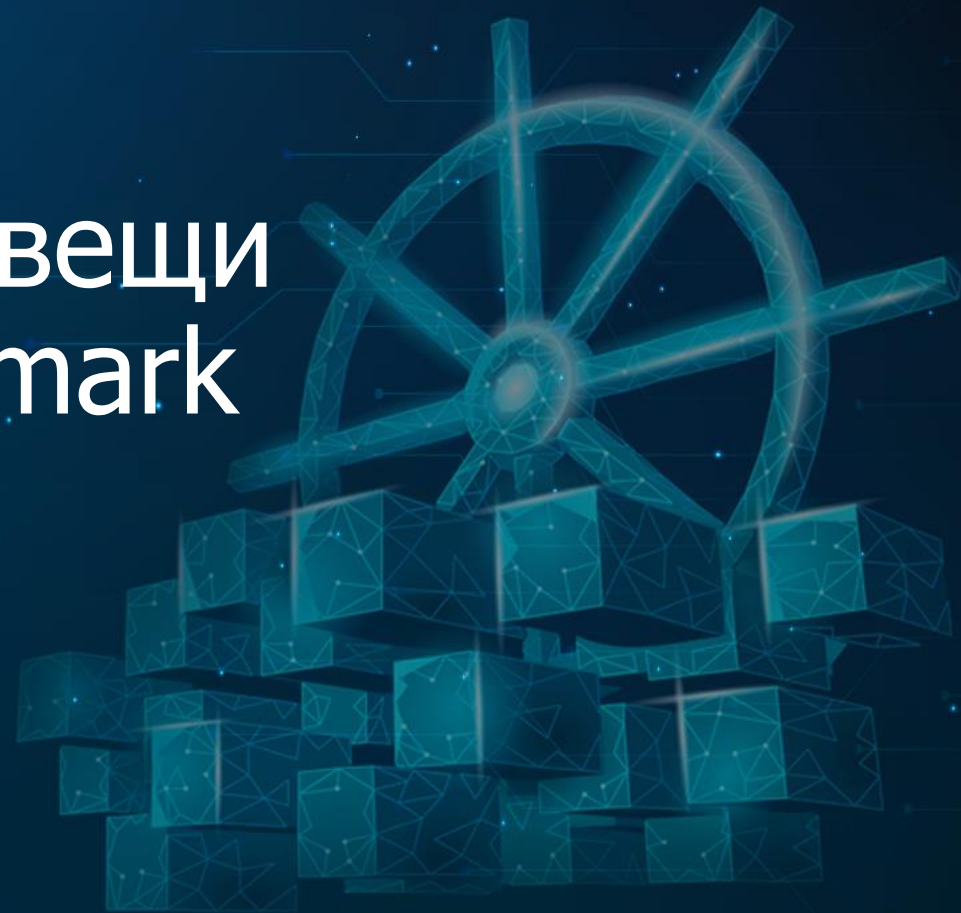
Стандарты безопасности для контейнерных сред

- CIS Kubernetes Benchmark
 - <https://www.cisecurity.org/benchmark/kubernetes/>
- NSA/CISA Kubernetes Hardening Guide
 - <https://www.cisa.gov/news-events/alerts/2022/03/15/updated-kubernetes-hardening-guide>
- Kubernetes Security Technical Implementation Guide (STIG)
 - <https://ncp.nist.gov/checklist/996>
- PCI Security Standards Council: Guidance for Containers and Container Orchestration Tools
 - <https://blog.pcisecuritystandards.org/new-information-supplement-guidance-for-containers-and-container-orchestration-tools>
- NIST Special Publication 800-190 "Application Container Security Guide"
 - <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf>
- Приказом ФСТЭК России №118. Требования по безопасности информации к средствам контейнеризации
 - <https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/trebovaniya-po-bezopasnosti-informatsii-utverzhdenny-prikazom-fstek-rossii-ot-4-iyulya-2022-g-n-118>
- ...

Материалы по теме

- [“Харденинг K8S. \(Не\)очевидные настройки”](#), Антон Гаврилов, PHDays 2022
- [“Стандарты безопасности в Kubernetes”](#), Константин Аксенов, VK Kubernetes Conf 2023

Очевидные и не очень вещи в CIS Kubernetes Benchmark



CIS Kubernetes Benchmark

- Компоненты Control Plane
 - Конфигурационные файлы узлов Control Plane
 - API Server
 - Controller Manager
 - Scheduler
- Проверка etcd
- Конфигурации Control Plane
 - Аутентификация и авторизация
 - Логирование/Аудит
- Worker Nodes
 - Конфигурационные файлы узлов Worker Nodes
 - Конфигурация kubelet
- Policies
 - RBAC и Service Accounts
 - Pod Security Standards
 - Network Policies и CNI
 - Управление Secrets
 - Сторонний Admission Control
 - Общие политики

№	Категория проверок	Количество*	Количество (%)
1	Компоненты Control Plane	60	46
2	Проверка etcd	7	5
3	Конфигурации Control Plane	5	4
4	Worker Nodes	23	18
5	Policies	35	27

* - в последней версии CIS Kubernetes Benchmark 1.8 всего 130 проверок

Интересные моменты

- Benchmark эволюционирует
 - 1.7 для 1.25, 131 проверка
 - 1.8 для 1.27, 130 проверок
- Нельзя автоматизировать:
 - 1 проверка для Windows систем
 - 5.2.11 Minimize the admission of Windows HostProcess Containers
 - 1 проверка требует данные за пределами Kubernetes
 - 5.1.7 Avoid use of system:masters group

Возможности Open Source решений



ПОДОПЫТНЫЕ

- kube-bench - Checks whether Kubernetes is deployed according to security best practices as defined in the CIS Kubernetes Benchmark
 - <https://github.com/aquasecurity/kube-bench>
- trivy-operator - Kubernetes-native security toolkit
 - <https://github.com/aquasecurity/trivy-operator>
- NeuVector - Full Lifecycle Container Security Platform
 - <https://github.com/neuvector/neuvector>
- StackRox - Kubernetes Security Platform
 - <https://github.com/stackrox/>

Результат работы

- Использовались самые последние версии решений на момент тестирования
 - Где-то отличались версии бенчмарков
- Сканирования проходили в одном кластере одновременно
 - Версия Kubernetes 1.24.17

Инструмент	Версия инструмента	Версия CIS	Описано проверок*	Выполнено проверок	Выполнено проверок (% от 130)
kube-bench	0.7.0	1.8	130	66	51
trivy-operator	0.18.0	1.8	116	108	83
StackRox	4.3.3	1.5	122	75	58
NeuVector	5.2.4-s1	1.6	88/31	???	???

* - в последней версии CIS Kubernetes Benchmark 1.8 всего 130 проверок

Kube-bench

- Классика =)
- Много что не автоматизировано

kube-bench / cfg / cis-1.8 / policies.yaml

mozillazg support CIS Kubernetes Benchmark v1.8.0 (#1527) ✓

Code

Blame

304 lines (270 loc) · 11.5 KB

```
1 ---
2 controls:
3 version: "cis-1.8"
4 id: 5
5 text: "Kubernetes Policies"
6 type: "policies"
7 groups:
8   - id: 5.1
9     text: "RBAC and Service Accounts"
10    checks:
11      - id: 5.1.1
12        text: "Ensure that the cluster-admin role is only granted to the cluster-admin user"
13        type: "manual"
14        remediation: |
15          Identify all clusterrolebindings to the cluster-admin role.
16          If they need this role or if they could use it, remove the binding.
17          Where possible, first bind users to a low-privileged role.
18          Remove the cluster-admin role from the clusterrolebinding to the cluster-admin user.
19          kubectl delete clusterrolebinding [name]
20        scored: false
21
22      - id: 5.1.2
23        text: "Minimize access to secrets (Manual)"
24        type: "manual"
25        remediation: |
26          Where possible, remove get, list and watch permissions for secrets.
27        scored: false
```

```
controls:
version: "cis-1.8"
id: 3
text: "Control Plane Configuration"
type: "controlplane"
groups:
  - id: 3.1
    text: "Authentication and Authorization"
    checks:
      - id: 3.1.1
        text: "Client certificate authentication should not be used for users (Manual)"
        type: "manual"
        remediation: |
          Alternative mechanisms provided by Kubernetes such as the use of OIDC should be implemented in place of client certificates.
        scored: false
      - id: 3.1.2
        text: "Service account token authentication should not be used for users (Manual)"
        type: "manual"
        remediation: |
          Alternative mechanisms provided by Kubernetes such as the use of OIDC should be implemented in place of service account tokens.
        scored: false
      - id: 3.1.3
        text: "Bootstrap token authentication should not be used for users (Manual)"
        type: "manual"
        remediation: |
          Alternative mechanisms provided by Kubernetes such as the use of OIDC should be implemented in place of bootstrap tokens.
```

trivy-operator

- REGO правила
 - Немного умеет в 5 раздел
- Не все проверки
- Некорректные правила

trivy-operator / deploy / helm / templates / specs / cis-1.23.yaml

Code

Blame

823 lines (823 loc) · 32.2 KB

```
769     severity: MEDIUM
770     - id: 5.3.2
771       name: Ensure that all Namespaces have Network Policies defined
772       description: Use network policies to isolate traffic in your cluster network
773       checks:
774         - id: AVD-KSV-0038
775           severity: MEDIUM
776     - id: 5.4.1
```

trivy-policies / checks / kubernetes / advanced / selector_usage_in_network_policies.rego

simar7 refactor(deps): Restructure into checks/

3de4cf5 · 2 months ago

Code

Blame

79 lines (67 loc) · 2.17 KB

Raw

Copy

Download

Edit

```
1  # METADATA
2  # title: "Selector usage in network policies"
3  # description: "ensure that network policies selectors are applied to pods or namespaces to restricted ingress and egress traffic within the pod network"
4  # scope: package
5  # schemas:
6  # - input: schema["kubernetes"]
7  # related_resources:
8  # - https://kubernetes.io/docs/tasks/administer-cluster/declare-network-policies/
9  # custom:
10 # id: KSV038
11 # avd_id: AVD-KSV-0038
12 # severity: MEDIUM
13 # short_code: selector-usage-in-network-policies
14 # recommended_action: "create network policies and ensure that pods are protected using the podSelector and/or the namespaceSelector options"
15 # input:
16 #   selector:
17 #     - type: kubernetes
18 package builtin.kubernetes.KSV038
```

```
deny[res] {
    lower(kubernetes.kind) == "networkpolicy"
    not hasSelector(input.spec)
    msg := "Network policy should uses podSelector and/or the namespaceSelector to restrict ingress and egress traffic within the Pod network"
    res := result.new(msg, input.spec)
}
```


NeuVector

- Очень странный ...

Kubernetes CIS Version: 1.6.0, Scanned at: Jan 31 2024 07:14:20

automated,category,description,level,profile,remediation,scored,tags,type,name

neuvector / agent / nvbench / kubernetes-cis-benchmark / cis-1.8.0 / master / 5_policies.yaml

Code Blame 485 lines (485 loc) · 24.3 KB

Raw Copy Download Edit View

```
3 title: "5 - Policies"
4 type: "master"
5 groups:
6   - id: 5.1
7     title: "5.1 - RBAC and Service Accounts"
8     checks:
9       - id: 5.1.1
10         description: Ensure that the cluster-admin role is only used where required (Manual)
11         type: master
12         category: kubernetes
13         scored: false
14         profile: Level 1
15         automated: false
16         tags: []
17         audit: |
18           check="$id - $description"
19           manual "$check"
20           manual " * Run kubectl get clusterrolebindings -o=custom-columns=NAME:.metadata.name
21         remediation: 'Identify all clusterrolebindings to the cluster-admin role. Check if they are u
22   - id: 5.1.2
23     description: Minimize access to secrets (Manual)
24     type: master
25     category: kubernetes
26     scored: false
27     profile: Level 1
28     automated: false
29     tags: []
30     audit: |
31       check="$id - $description"
32       manual "$check"
33       manual " * Review the users who have get, list or watch access to secrets objects in the Kubernetes API."
```

Filtered: 31 / 33



Advanced Filter

Filter

Category	Name	Status	Scored	Profile	Impact	CSV
kubernetes	K.1.2.1	WARN	N	Level 1	1	Download
kubernetes	K.1.2.10	WARN	N	Level 1	1	Download
kubernetes	K.1.2.12	WARN	N	Level 1	1	Download
kubernetes	K.1.2.13	WARN	N	Level 1	1	Download
kubernetes	K.1.2.16	WARN	Y	Level 1	1	Download
kubernetes	K.1.2.19	WARN	Y	Level 1	1	Download

- Много что не автоматизировано
- Зброшен
- У RedHat есть свой [compliance-operator](#)

stackrox / pkg / compliance / checks / standards / cis_kubernetes.go

josephaltmaier ROX-5187 incorporate Boo's fixes (stackrox/rox#6270)

f1c7917 4 years ago

Code Blame 9 lines (7 loc) · 284 Bytes

```
1 package standards
2
3 // CISKubernetes is the string name of this standard
4 const CISKubernetes = "CIS_Kubernetes_v1_5"
5
6 // CISKubeCheckName takes a check ID and returns a fo
7 func CISKubeCheckName(checkName string) string {
8     return CheckName(CISKubernetes, checkName)
9 }
```

CIS Kubernetes v1.5
Standard



Standard: x

CIS Kubernetes v1.5 x

122 controls across 1 cluster

pkg/compliance/checks/pcidss32/check713/check.go

```
12 }
13
14 func clusterIsCompliant() *standards.CheckAndMetadata {
15     // This is a partial check. The evidence from this check will be folded together with evidence generated in central
16     checkAndMetadata := common.MasterAPIServerRBACConfigurationCommandLine()
17     checkAndMetadata.Metadata.InterpretationText = interpretationText
18 }
```

Show

5.1.2 - Minimize access to secrets	0%
5.1.3 - Minimize wildcard use in Roles and ClusterRoles	0%
5.1.4 - Minimize access to create pods	0%
5.1.5 - Ensure that default service accounts are not actively used	0%

Control guidance

The following property cannot be checked automatically by StackRox, and thus must be ensured manually: Minimize access to secrets

```
12 }
13
14 func clusterIsCompliant() *standards.CheckAndMetadata {
15     // This is a partial check. The evidence from this check will be folded together with evidence generated in central
16     checkAndMetadata := common.MasterAPIServerRBACConfigurationCommandLine()
17     checkAndMetadata.Metadata.InterpretationText = interpretationText
18     return checkAndMetadata
19 }
```

Ахиллесова пята

- 5 раздел CIS Kubernetes benchmark слабое место всех инструментов
 - Это 27% от всех проверок

5 Policies	215
5.1 RBAC and Service Accounts	216
5.1.1 Ensure that the cluster-admin role is only used where required (Manual)	217
5.1.2 Minimize access to secrets (Manual)	219
5.1.3 Minimize wildcard use in Roles and ClusterRoles (Manual)	221
5.1.4 Minimize access to create pods (Manual)	223
5.1.5 Ensure that default service accounts are not actively used. (Manual)	225
5.1.6 Ensure that Service Account Tokens are only mounted where necessary (Manual)	227
5.1.7 Avoid use of system:masters group (Manual)	229
5.1.8 Limit use of the Bind, Impersonate and Escalate permissions in the Kubernetes cluster (Manual)	231
5.1.9 Minimize access to create persistent volumes (Manual)	233
5.1.10 Minimize access to the proxy sub-resource of nodes (Manual)	234
5.1.11 Minimize access to the approval sub-resource of certificatesigningrequests objects (Manual)	236
5.1.12 Minimize access to webhook configuration objects (Manual)	238
5.1.13 Minimize access to the service account token creation (Manual)	240
5.2 Pod Security Standards	241
5.2.1 Ensure that the cluster has at least one active policy control mechanism in place (Manual)	242
5.2.2 Minimize the admission of privileged containers (Manual)	243
5.2.3 Minimize the admission of containers wishing to share the host process ID namespace (Manual)	245
5.2.4 Minimize the admission of containers wishing to share the host IPC namespace (Manual)	247
5.2.5 Minimize the admission of containers wishing to share the host network namespace (Manual)	249
5.2.6 Minimize the admission of containers with allowPrivilegeEscalation (Manual)	251
5.2.7 Minimize the admission of root containers (Manual)	253
5.2.8 Minimize the admission of containers with the NET_RAW capability (Manual)	255
5.2.9 Minimize the admission of containers with added capabilities (Manual)	257
5.2.10 Minimize the admission of containers with capabilities assigned (Manual)	259
5.2.11 Minimize the admission of Windows HostProcess Containers (Manual)	261
5.2.12 Minimize the admission of HostPath volumes (Manual)	263
5.2.13 Minimize the admission of containers which use HostPorts (Manual)	264
5.3 Network Policies and CNI	265
5.3.1 Ensure that the CNI in use supports Network Policies (Manual)	266
5.3.2 Ensure that all Namespaces have Network Policies defined (Manual)	268
5.4 Secrets Management	270
5.4.1 Prefer using secrets as files over secrets as environment variables (Manual)	271
5.4.2 Consider external secret storage (Manual)	273
5.5 Extensible Admission Control	274
5.5.1 Configure Image Provenance using ImagePolicyWebhook admission controller (Manual)	275
5.7 General Policies	277
5.7.1 Create administrative boundaries between resources using namespaces (Manual)	278
5.7.2 Ensure that the seccomp profile is set to docker/default in your pod definitions (Manual)	280
5.7.3 Apply Security Context to Your Pods and Containers (Manual)	282
5.7.4 The default namespace should not be used (Manual)	284

По итогу о состоянии OpenSource реализаций

1. Обрезанный Benchmark
 - Не все пункты описаны
2. Отсутствие реализации проверок
 - Идет как TODO или manual check
3. Не полная реализация проверок
 - Этот факт даже отмечается в комментариях (partial check)
4. Некорректная реализация пунктов
 - Логика проверки не отражает исходный пункт
5. Отсутствие поддержки новых версий Kubernetes
 - Не забывайте, что CIS Kubernetes Benchmark обновляется

Как результат получаем не корректную картину по своей инфраструктуре!

Не ленитесь читать код OpenSource проектов!

Взгляд Luntry



Compliance

Cluster Name: Gatewayer

Group by subsections: 14.02.2024 13:00 cis:1.8 Select date range Export 1 Checks: 149 Statuses: 4 Severities: 4

14.02.2024 11:00:05 14.02.2024 13:00:00

62.04% 61.63%

3. Control Plane Configuration 5 Controls (0 Pass) (0 Fail) (0 Error) (0 Warn)

4. Worker Nodes 23 Controls (0 Pass) (0 Fail) (0 Error) (0 Warn)

5. Policies 35 Controls (-1 Pass) (+1 Fail) (0 Error) (0 Warn)

5.1. RBAC and Service Accounts 13 Controls (0 Pass) (0 Fail) (0 Error) (0 Warn)

5.2. Pod Security Standards 13 Controls (-1 Pass) (+1 Fail) (0 Error) (0 Warn)

ID	Status	Severity	Name	Before	After
5.2.1	=	HIGH	Ensure that the cluster has at least one active policy control mechanism in place	0	0
5.2.2	=	HIGH	Minimize the admission of privileged containers	0	0
5.2.3	=	MEDIUM	Minimize the admission of containers wishing to share the host process ID namespace	-1	+2
5.2.4	↓	MEDIUM	Minimize the admission of containers wishing to share the host IPC namespace	0	+1
5.2.5	=	MEDIUM	Minimize the admission of containers wishing to share the host network namespace	0	0
5.2.6	=	MEDIUM	Minimize the admission of containers with allowPrivilegeEscalation	-1	+7
5.2.7	=	MEDIUM	Minimize the admission of root containers	0	0

Сравнение

Инструмент	Версия CIS	Описано проверок*	Выполнено проверок	Выполнено проверок (% от 130)
kube-bench	1.8	130	66	51
trivy-operator	1.8	116	108	83
StackRox	1.5	122	75	58
NeuVector	1.6	88/31	???	???
Luntry	1.8	130	128	99

* - в последней версии CIS Kubernetes Benchmark 1.8 всего 130 проверок

Возможности Luntry Compliance

- 99% автоматизации стандарта CIS Kubernetes Benchmark
- Отсутствие дополнительной нагрузки на API Server
- Создание собственных шаблонов
- Гибкая фильтрация результатов
- Экспорт результатов в PDF
- Сканирование по расписанию
- Отслеживание изменений/прогресса/регресса
- Возможность сравнение результатов сканирований

Дорожная карта развития

- Принятие рисков и закрытие через отдельные митигейшны
- Возможность создавать собственные compliance

ИТОГ

1. Хотя бы раз полностью ознакомиться с CIS Kubernetes Benchmark
2. Перепроверяйте любой OpenSource ;)
3. CIS Kubernetes Benchmark можно почти полностью автоматизировать!
4. Соответствие CIS Kubernetes Benchmark может быть хорошим стартом для обеспечения безопасности кластеры

Спасибо за внимание!

Дмитрий Евдокимов
Founder&CTO



Email: de@luntry.ru



Twitter: @evdokimovds
@Qu3b3c



Channel: @k8security



Site: www.luntry.ru



 [k8security](#)    [luntrysolution](#)