SOAR в Kubernetes малой кровью

Дмитрий Евдокимов
Luntry

Agenda

- Введение в SOAR
- Kubernetes Policy Management
  - (Cluster) PolicyReport
- SOAR в Kubernetes
  - Deploy-фаза
  - Playbooks
  - Runtime-фаза
  - Playbooks
- Заключение

ВНИМАНИЕ!

Все трюки выполнены профессионалами !
Во время съёмок никто не пострадал!

Не пытайтесь это повторить в домашних условиях!

WhoAmI

- Основатель и технический директор Luntry
- Опыт в ИБ более 10 лет
- Соорганизатор конференций ZeroNights, DEFCON Russia (#7812)
- 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.
- Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, PHDays, DevOpsConf, KuberConf, VK Kubernetes Conference, HighLoad++ и др.

3 часть - заключительная

1 часть

- DevOpsConf 2021
  - Kubernetes: трансформация к SecDevSecOpsSec

2 часть

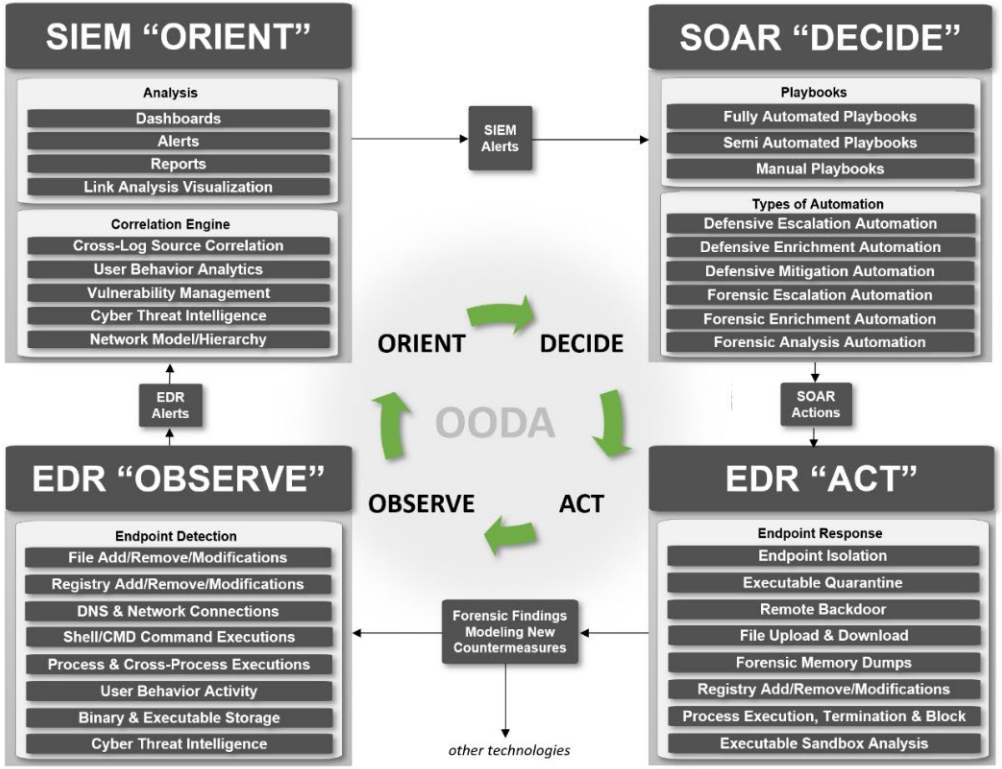
- VK Kubernetes Conference 2021
  - Kubernetes Resource Model (KRM): Everything-as-Code

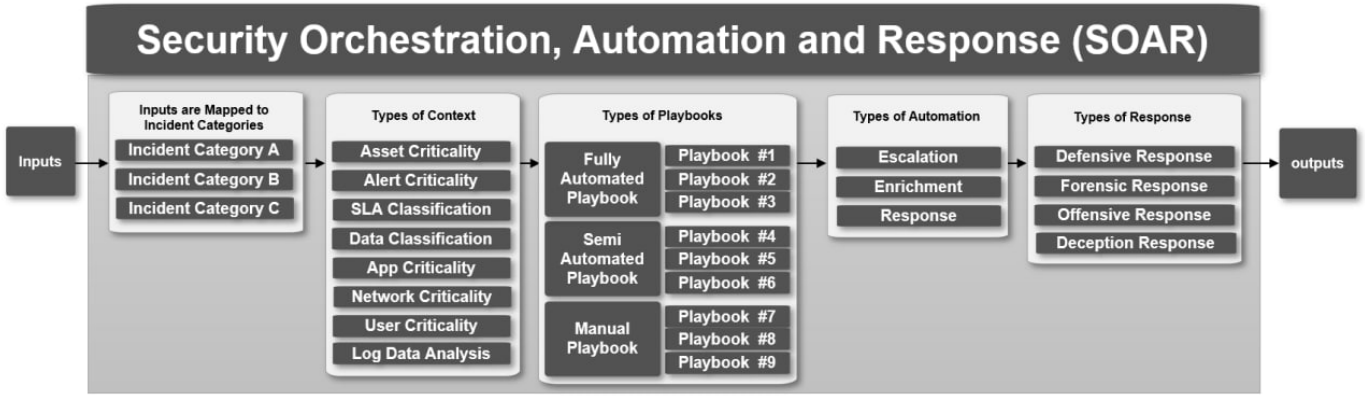
3 часть

- Welcome ;)

Введение в SOAR

SOAR





"An OODA-driven SOC Strategy using: SIEM, SOAR and EDR"

DoD Zero Trust Reference Architecture

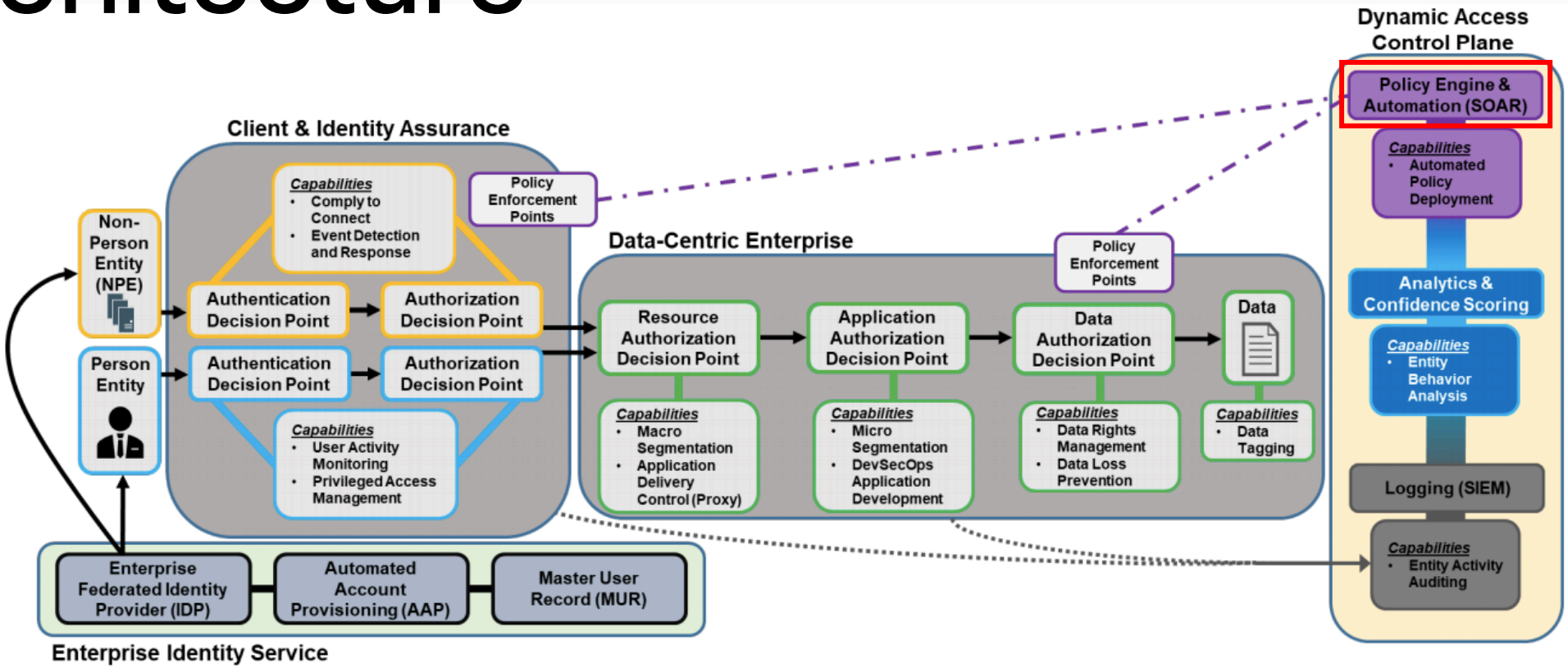


Figure 2: High-Level Operational Concept (OV-1)

[“Department of Defense (DOD) - Zero Trust Reference Architecture”]

DoD Enterprise DevSecOps Reference Design

Table 15: CSP Managed Service Monitoring Tools

Tool	Features	Benefits	Baseline
Netflow Analysis	Logs network traffic within as enclave Network troubleshooting	Helps to find anomalous patterns across environment and Platform	REQUIRED
Centralized Logging	Stores logs from the entire environment. Used by the SIEM/SOAR for log analysis and incident detection	Place to store logs across environment and Platform	REQUIRED
Centralized Analysis	SIEM/SOAR for log analysis and incident detection Tier 3 CSSP tools	Helps to find anomalous patterns across environment and Platform	RECOMMENDED

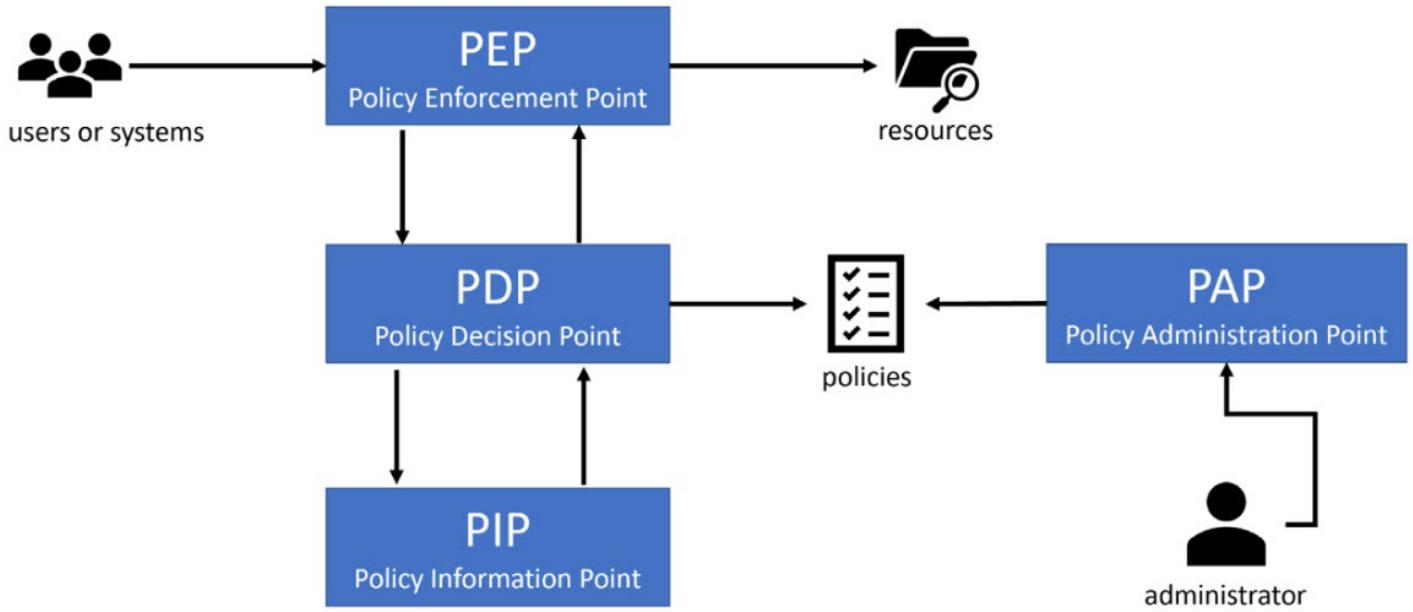
5.2.1 CSP Managed Services for Continuous Monitoring

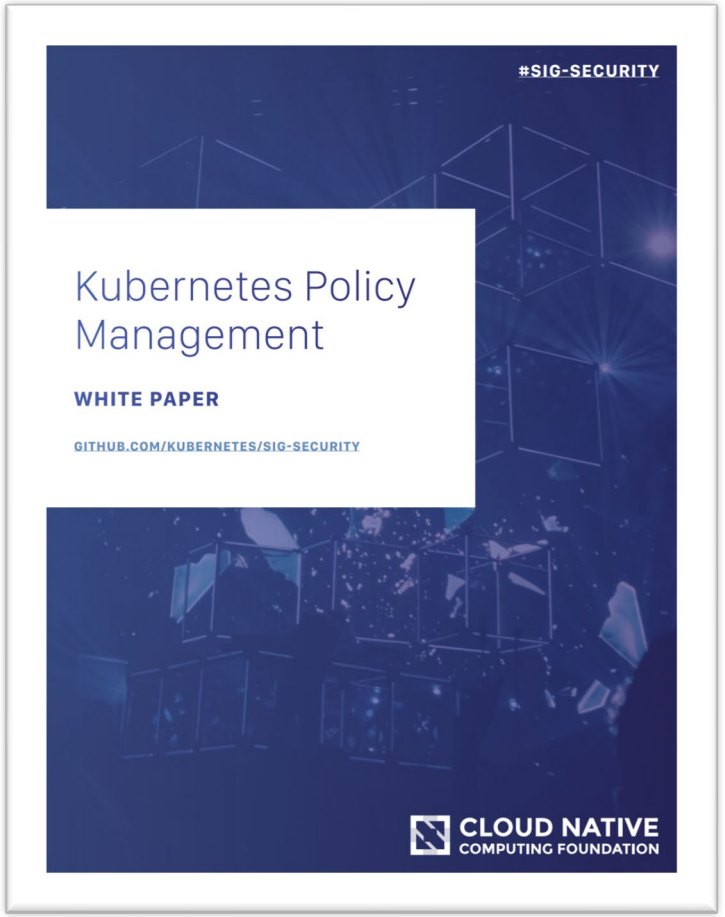
The use of CSP managed services for monitoring alongside 3rd party security tools should always be viewed through a "both/and" lens instead of an "either/or" lens. CSP managed services can be utilized to monitor CSP resources & services, netflow, and entity behavior analysis at a deeper level than with 3rd party tools alone. It may also be possible to employ CSP managed services to perform log analysis (SIEM/SOAR). The monitoring ecosystem should rely on curated IaC to instantiate the monitored environment to the maximum extent possible, ensuring completeness and accelerating the A&A process.

[DoD Enterprise DevSecOps Reference Design: CNCF Kubernetes]

Kubernetes Policy Management

Kubernetes Policy Management Paper





[“Kubernetes Policy Management Paper”]

PolicyReport & ClusterPolicyReport

Специализированные k8s-ресурсы для отчетов.

Способы применения:

- Security-as-Code
- Policy-as-Code
- Compliance-as-Code
- Detection-as-Code
- ...

```yaml
- category: Best Practices
  message: validation rule 'autogen-validate-image-tag' passed.
  policy: disallow-latest-tag
  resources:
  - apiVersion: apps/v1
    kind: Deployment
    name: adservice
    namespace: boutique
    uid: 7f5feb5b-76ca-4611-85d5-d57440eb4c0c
  result: pass
  rule: autogen-validate-image-tag
  scored: true
  severity: medium
  source: Kyverno
  timestamp:
    nanos: 0
    seconds: 1654803230
summary:
  error: 0
  fail: 0
  pass: 24
  skip: 0
  warn: 0
```

Примеры PolicyReport

```yaml
apiVersion: wgpolicyk8s.io/v1alpha2
kind: PolicyReport
metadata:
  creationTimestamp: "2022-06-02T11:21:30Z"
  generation: 4
  name: falco-policy-report-956c41f3
  namespace: test
  resourceVersion: "597598"
  uid: cd9ad944-7984-40fe-9301-b8732cf4f4cb
results:
- category: SI - System and Information Integrity
  message: '11:21:28.059042000: Informational Privileged container started (user=<NA>
    user_loginuid=0 command=container:0622b6d8a2a8 k8s.ns=test k8s.pod=sensor-lshhm
    container=0622b6d8a2a8 image=registry.com/sensor:latest) k8s.ns=test
    k8s.pod=sensor-lshhm container=0622b6d8a2a8'
  policy: Launch Privileged Container
  properties:
    container.id: 0622b6d8a2a8
    container.image.repository: registry.com/sensor
    container.image.tag: latest
    evt.time: "1654168888059042000"
    k8s.ns.name: test
    k8s.pod.name: sensor-lshhm
    proc.cmdline: container:0622b6d8a2a8
    user.loginuid: "0"
    user.name: <nil>
  result: fail
  severity: high
  source: Falco
  timestamp:
    nanos: 59042000
    seconds: 28
```

```yaml
results:
- category: Vulnerability Scan
  message: 'coreutils: Non-privileged session can escape
      to the parent session in chroot'
  policy: CVE-2016-2781
  properties:
    artifact.repository: src/collector
    artifact.tag: latest
    installedVersion: 8.32-4.1ubuntu1
    primaryLink: https://avd.aquasec.com/nvd/cve-2016-2781
    registry.server: registry.com
    resource: coreutils
    resultID: 3569360dd9650609ebaf401f36a62ba4b62397e4
    score: "8.6"
  resources:
  - apiVersion: apps/v1
    kind: ReplicaSet
    name: collector-cdf444fdc
    namespace: test
    uid: 4ca4723c-790e-47ee-8b9c-7e96ec016b46
  result: warn
  severity: low
  source: Trivy Vulnerability
  timestamp:
    nanos: 0
    seconds: 1654002075
```

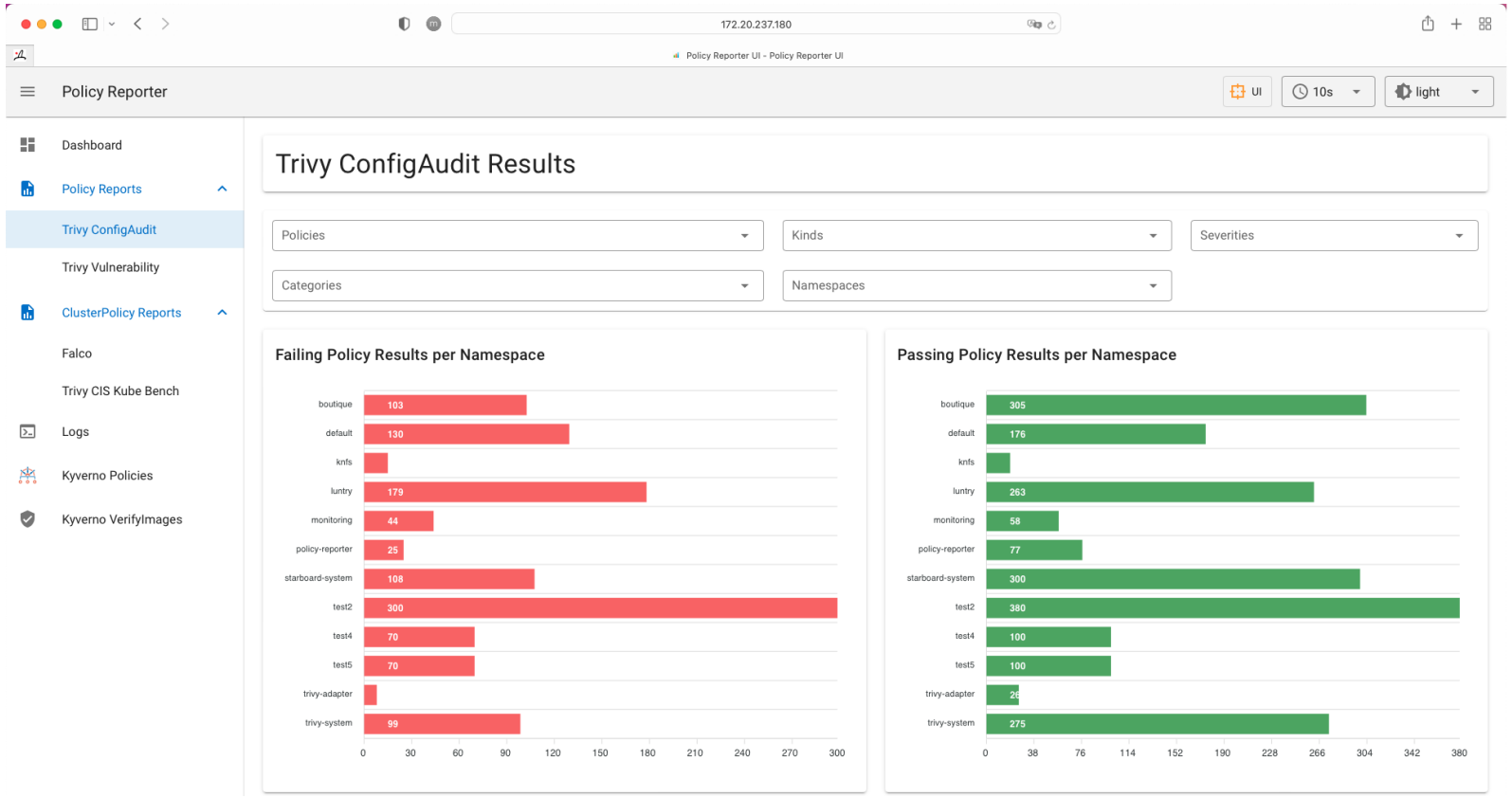
Работа с PolicyReports

- Policy Reporter - Monitoring and Observability Tool for the PolicyReport CRD with an optional UI
- Kyverno - Kubernetes Native Policy Management
- Falco adapter - Falco Policy Report adapter receives Falco events and produces one or more Policy Reports.
- kube-bench adapter - Building a prototype of Policy Report Generator. It aims to run a CIS benchmark check like kube-bench and produce a policy report.
- kubearmor-adapter - This KubeArmor Policy Report adapter converts output received from KubeArmor and produces a policy report based on the Policy Report Custom Resource Definition.
- Trivy Operator PolicyReport Adapter - Creates PolicyReports based on the different Trivy Operator CRDs like VulnerabilityReports

Tool: Policy Reporter

Есть поддержка:

- Grafana Loki
- Elasticsearch
- Slack
- Discord
- MS Teams
- S3,
- Policy Reporter UI



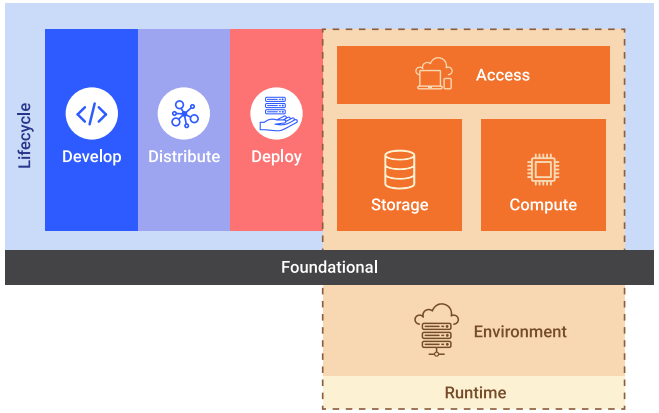
Недостатки Policy Reports

- Отсутствие универсальности
  - Не все результаты хорошо ложатся на данный ресурс
- Проблема разграничения доступа к ресурсу
  - Доступ в RBAC к (Cluster) PolicyReport типу дает доступ ко всем отчетам

```
kubectl get polr
NAME                                                  PASS   FAIL   WARN   ERROR   SKIP   AGE
falco-policy-report-2554a679                          0      8      0      0       0      84m
falco-policy-report-5653071e                          0      6      0      0       0      77m
trivy-audit-polr-access                               19     15     0      0       0      51m
trivy-vuln-polr-everything-allowed-exec-pod           0      0      16     0       0      51m
trivy-audit-polr-nginx-6799fc88d8                     20     14     0      0       0      50m
trivy-audit-polr-ubuntu-deployment-56bb7c4678         20     14     0      0       0      51m
trivy-audit-polr-webadmin-74bb488d96                  20     14     0      0       0      50m
```

SOAR в Kubernetes

2 точки применения



Deploy-фаза

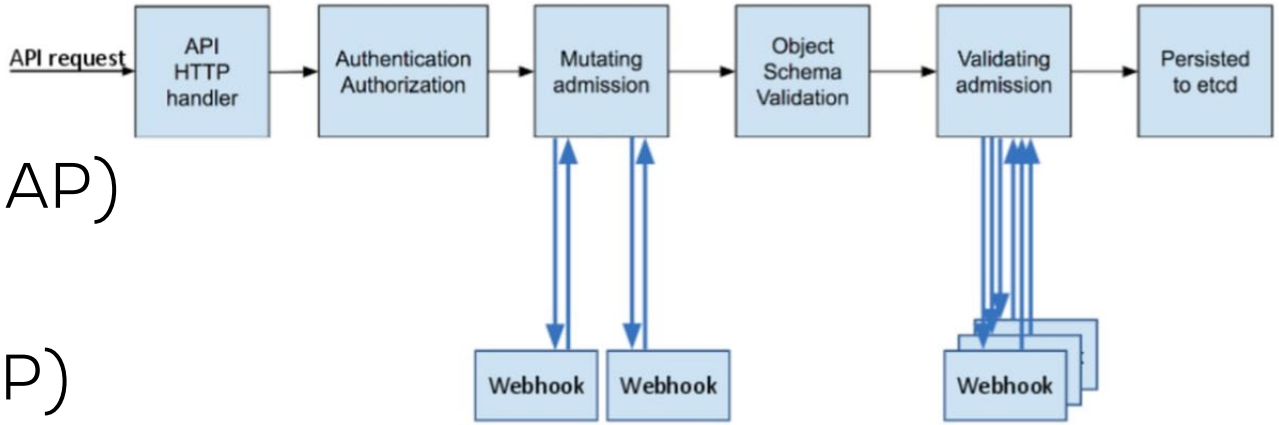
- Admission controllers
  - Policy Engines
    - Kyverno, OPA Gatekeeper, JSPolicy, Kubewarden

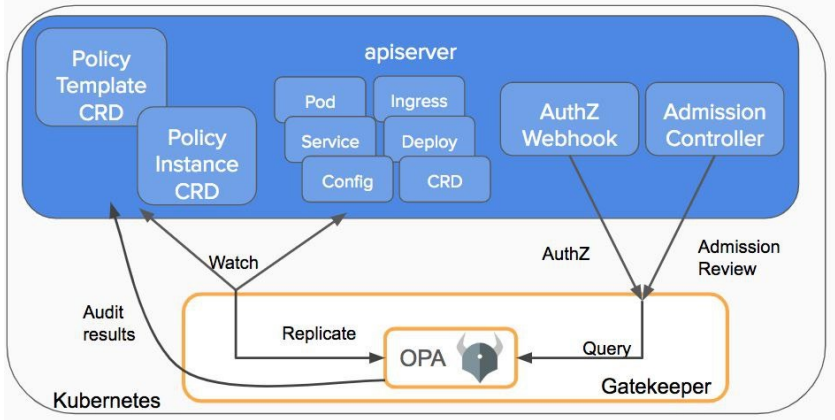
Runtime-фаза

- Agent
  - Luntry, Falco, Tracee, Tetragon, KubeArmor
- Response Engine
  - Argo Events + Argo Workflow, falcosidekick

Deploy-фаза

- Policy Administration Point (PAP)
  - Policy Engine
- Policy Enforcement Point (PEP)
  - Admissions
- Policy Decision Point (PDP)
  - Policy Engine
- Policy Information Point (PIP)
  - Любая сторонняя система





Playbook 1: Запрет на exec в Pod

- Ситуация:
  - Атакующий скомпрометировал учетную запись и пытается получить shell или выполнить команду в контейнере
- Возможна валидация любых Kubernetes resources и subresources
  - Разрешить
  - Запретить

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: deny-exec-by-pod-and-container
  annotations:
    policies.kyverno.io/title: Block Pod Exec by Pod and Container
    policies.kyverno.io/category: Sample
    policies.kyverno.io/minversion: 1.4.2
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      The `exec` command may be used to gain shell access, or run other commands, in a Pod's container. While this can
      be useful for troubleshooting purposes, it could represent an attack vector and is discouraged.
      This policy blocks Pod exec commands to containers named `nginx` in Pods starting
      with name `myapp-maintenance`.
spec:
  validationFailureAction: enforce
  background: false
  rules:
  - name: deny-nginx-exec-in-myapp-maintenance
    match:
      resources:
        kinds:
        - PodExecOptions
    preconditions:
      all:
      - key: "{{ request.operation }}"
        operator: Equals
        value: CONNECT
      - key: "{{ request.name }}"
        operator: Equals
        value: myapp-maintenance*
    validate:
      message: Nginx containers inside myapp-maintanence Pods may not be exec'd into.
      deny:
        conditions:
          all:
          - key: "{{ request.object.container }}"
            operator: Equals
            value: nginx
```

Playbook 2: Добавление securityContext

- Ситуация:
  - Разработчик в обход установленных pipeline пытается выкатить небезопасное приложение
- Возможно мутировать любые Kubernetes resources

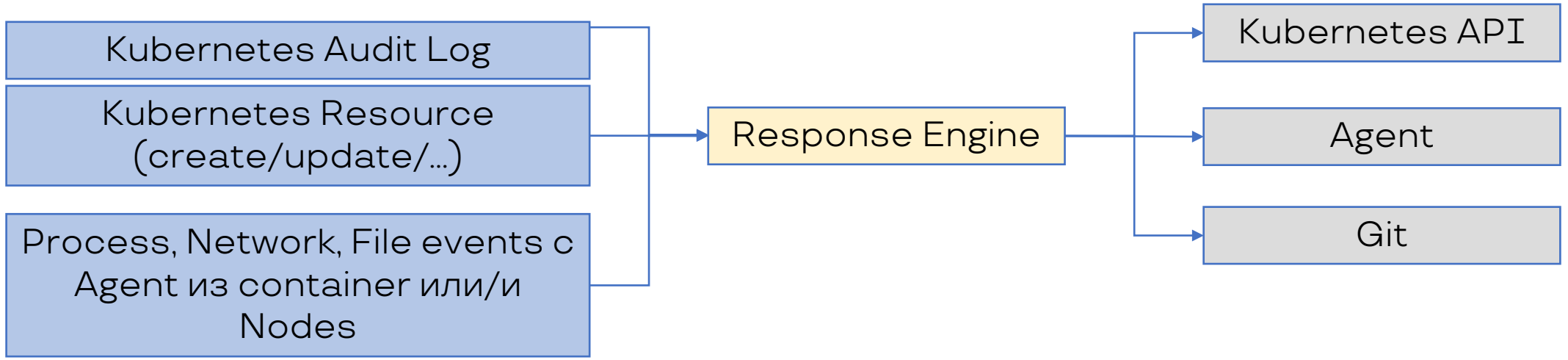
```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: apply-pss-restricted-profile
  annotations:
    policies.kyverno.io/title: Apply PSS Restricted Profile
    policies.kyverno.io/category: Other
    kyverno.io/kyverno-version: 1.6.2
    kyverno.io/kubernetes-version: "1.23"
    policies.kyverno.io/subject: Pod
    policies.kyverno.io/description: >-
      Pod Security Standards define the fields and their options which
      are allowable for Pods to achieve certain security best practices. While
      these are typically validation policies, workloads will either be accepted or
      rejected based upon what has already been defined. It is also possible to mutate
      incoming Pods to achieve the desired PSS level rather than reject. This policy
      sets all the fields necessary to pass the PSS Restricted profile.
spec:
  rules:
  - name: add-pss-fields
    match:
      any:
      - resources:
          kinds:
          - Pod
    mutate:
      patchStrategicMerge:
        spec:
          securityContext:
            seccompProfile:
              type: RuntimeDefault
            runAsNonRoot: true
            runAsUser: 1000
            runAsGroup: 3000
            fsGroup: 2000
          containers:
          - (name): "?*"
            securityContext:
              privileged: false
              capabilities:
                drop:
                - ALL
              allowPrivilegeEscalation: false
```

Playbook 3: Создание запрещающей NetworkPolicy

- Ситуация:
  - Атакующий пытается создать новый namespace без ограничений и через него развивать атаку
- Возможно генерировать любые Kubernetes resources

```yaml
apiVersion: kyverno.io/v1
kind: ClusterPolicy
metadata:
  name: default
spec:
  rules:
  - name: deny-all-traffic
    match:
      any:
      - resources:
          kinds:
          - Namespace
    exclude:
      any:
      - resources:
          namespaces:
          - kube-system
          - default
          - kube-public
          - kyverno
    generate:
      kind: NetworkPolicy
      apiVersion: networking.k8s.io/v1
      name: deny-all-traffic
      namespace: "{{request.object.metadata.name}}"
      data:
        spec:
          # select all pods in the namespace
          podSelector: {}
          policyTypes:
          - Ingress
          - Egress
```

Runtime-фаза

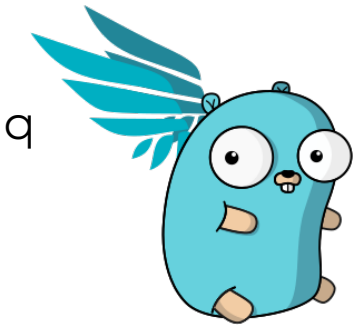


- Policy Administration Point (PAP)
  - Response Engine
- Policy Enforcement Point (PEP)
  - Kubernetes API, Agent, Git
- Policy Decision Point (PDP)
  - Response Engine
- Policy Information Point (PIP)
  - Любая сторонняя система

Tool: falcosidekick

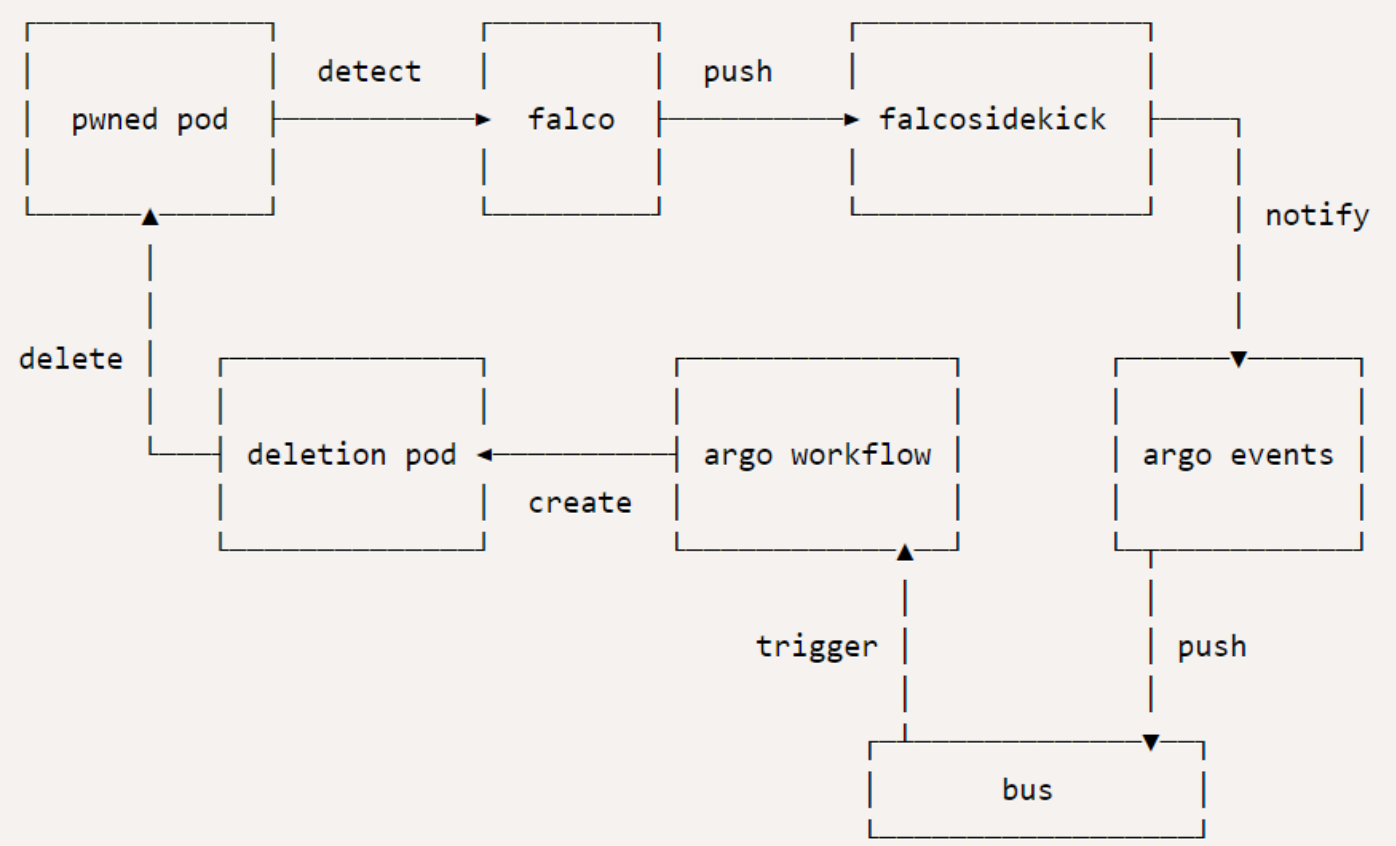
Event source: Сообщения в определённом формате (Falco, Tracee)

Triggers:



- Chat: Slack, Rocketchat, Mattermost, Teams, Discord, Google Chat, Zoho Cliq
- Metrics/Observability: Datadog, Influxdb, Prometheus, Wavefront
- Alerting: AlertManager, Opsgenie, PagerDuty
- Logs: Elasticsearch, Loki, AWS CloudWatchLogs, Grafana, Syslog
- Object Storage: AWS S3, GCP Storage, Yandex S3 Storage
- FaaS/Serverless: AWS Lambda, Kubeless, OpenFaaS, Knative, GCP Cloud Run, GCP Cloud Functions, Fission
- Message queue/Streaming: NATS, STAN (NATS Streaming), AWS SQS, AWS SNS, AWS Kinesis, GCP PubSub, Apache Kafka, Kafka, Rest Proxy, RabbitMQ, Azure Event Hubs
- Email: SMTP
- Web: Webhook, WebUI
- Other: Tekton, Flux v2, **Argo Events + Argo Workflow**, **Policy Report**

Пример: falcosidekick + Argo



[Kubernetes Response Engine, Part 5: Falcosidekick + Argo]

Tool: Argo Events + Argo Workflow

Event Sources: AMQP, AWS SNS, AWS SQS, Azure Events Hub, Bitbucket, Bitbucket Server, Calendar, Emitter, File Based Events, GCP PubSub, Generic EventSource, GitHub, GitLab, HDFS, **K8s Resources**, Kafka, Minio, NATS, NetApp StorageGrid, MQTT, NSQ, Pulsar, Redis, Slack, Stripe, **Webhooks**

Triggers: **Argo Workflows**, **Standard K8s Objects**, HTTP Requests / Serverless Workloads (OpenFaaS, Kubeless, KNative etc.), AWS Lambda, NATS Messages, Kafka Messages, Slack Notifications, Azure Event Hubs Messages, Argo Rollouts, Custom Trigger / Build Your Own Trigger, Apache OpenWhisk, Log Trigger

Argo Events - The Event-Based Dependency Manager for Kubernetes

Argo Workflows - The workflow engine for Kubernetes

Incident Response Workflow

Общий алгоритм

Создаем Event Source

```
apiVersion: argoproj.io/v1alpha1
kind: EventSource
```

Складывает события в Event Bus

Создаем Sensor

Создаем Service Account

Деплоим Sensor

```
apiVersion: argoproj.io/v1alpha1
kind: Sensor
```

В Sensor указываем Workflow со ссылкой на WorkflowTemplate

Создаем Workflow Template

Создаем Service Account*

Деплоим WorkflowTemplate

```
apiVersion: argoproj.io/v1alpha1
kind: WorkflowTemplate
```

В WorkflowTemplate указываем образ с полезной нагрузкой

Playbook 4: Изоляция Pod

Ситуация:

- Внутри Pod обнаружена вредоносная, аномальная активность
- Необходимо его изолировать по сети, для остановки сетевой активности (сканирований, общения с C&C и т.д.)
- Необходимо провести последующий анализ инцидента

Playbook 4: Изоляция Pod (1/2)

Вариант 1: Если `NetworkPolicy` не применяются в компании.

1. Создаем `NetworkPolicy` политику
   - Можно заранее
2. Ставим `label` на `Pod`
   - `status: compromised`

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: isolate-compromised-pod
spec:
  podSelector:
    matchLabels:
      status: compromised
  policyTypes:
  - Ingress
  - Egress
```

"NetworkPolicy — родной межсетевой экран Kubernetes"

Playbook 4: Изоляция Pod (2/2)

Вариант 2: Если `NetworkPolicy` уже применяются ?!

1. Создаем `deny` политики от `Custom NetworkPolicy` (Calico/Cilium)
   - Можно заранее
2. Ставим `label` на `Pod`
   - `status: compromised`

```
apiVersion: projectcalico.org/v3
kind: GlobalNetworkPolicy
metadata:
  name: isolate-compromised-pod
spec:
  order: 10
  selector: status == 'compromised'
  ingress:
  - action: Deny
    protocol: TCP
    source:
      selector: all()
  egress:
  - action: Deny
    protocol: TCP
    source:
      selector: all()
```

```
apiVersion: "cilium.io/v2"
kind: CiliumClusterwideNetworkPolicy
metadata:
  name: "isolate-compromised-pod"
spec:
  endpointSelector:
    matchLabels:
      status: compromised
  egressDeny:
  - toEntities:
    - "all"
  ingressDeny:
  - fromEntities:
    - "all"
```

Playbook 5: Временное усиление изоляции container

Сценарий:

- В `PolicyReport/VulnerabilityReport/...` содержится High critical RCE уязвимость
- Заново выкатываем данное приложение с установленным значением `runtimeClassName`, где используется sandbox или microVM


```yaml
kind: RuntimeClass
apiVersion: node.k8s.io/v1beta1
metadata:
    name: native
spec:
    runtimeHandler: runc
---
kind: RuntimeClass
apiVersion: node.k8s.io/v1beta1
metadata:
    name: gvisor
spec:
    runtimeHandler: gvisor
----
kind: RuntimeClass
apiVersion: node.k8s.io/v1beta1
metadata:
    name: kata-containers
spec:
    runtimeHandler: kata-containers
----

kind: RuntimeClass
apiVersion: node.k8s.io/v1beta1
metadata:
    name: sandboxed
spec:
    runtimeHandler: gvisor
```

```yaml
apiVersion: v1
kind: Pod
metadata:
    name: mypod
spec:
    runtimeClassName: gvisor
```

Playbook 6: Отзыв Service Account token

Сценарий:

- Обращение к `SelfSubjectAccessReview` или `SelfSubjectRulesReview` APIs от `service accounts` или `nodes`
  - Это значит, что атакующий пытается понять какими правами он обладает
- Создаем новый `Service Account` и добавляем его вместо скомпрометированного
- Удаляем скомпрометированный `Service Account`

Service Account Tokens

A service account is an automatically enabled authenticator that uses signed bearer tokens to verify requests. The plugin takes two optional flags:

- `--service-account-key-file` A file containing a PEM encoded key for signing bearer tokens. If unspecified, the API server's TLS private key will be used.
- `--service-account-lookup` If enabled, tokens which are deleted from the API will be revoked.

Link

Rotating Kubernetes service account credentials

If a Kubernetes service account credential is compromised and you wish to revoke the compromised credentials, take one of the following approaches:

- Create a new Kubernetes service account, migrate the Pod and any authorization to the new service account, and then revoke access to the old Kubernetes service account.

Link

Согласованность данных

Важный момент, чтобы не сломать Prod ;)

- Необходимо иметь и сохранять единый источник правды

Варианты:

- Писать напрямую в Kubernetes API или команда на Agent
  - Ломает согласованность данных =(
- Писать в Git
  - GitOps
  - Использовать специальный быстрый security pipeline

Дискуссия по данному вопросу.

Заключение

Выводы

- От self-healing до self-defence
- Argo Events + Argo Workflow отлично подходят для реагирования
- SOAR в Kubernetes по силам любому

Спасибо за внимание!

Email: de@luntry.ru

Twitter: @evdokimovds

Telegram: @Qu3b3c

LUNTRY

luntry.ru