Falcoルール

Kubernetes Audit

https://github.com/falcosecurity/falco/blob/master/rules/k8s_audit_rules.yaml

2020年12月16日 時点

本文の内容は、Cloud Native Runtime Security であるFalco (https://github.com/falcosecurity/falco/）において、2020年12月16日時点を元に作成したドキュメントです。

はじめに

このドキュメントは、CNCFでデファクトスタンダードとなっているクラウドネイティブランタイムセキュリティの仕組みである Falcoにおける Kubernetes Auditをソースとした Falcoルール個々の内容について記述しています。Falcoルールには、システムコールをソースとしたルール、Kubernetes Auditをソースとしたルール、アプリケーションに特化したルールがあります。また、Cloud Native Security Hub(https://securityhub.dev/）でもFalcoルールは公開されています。記述している、マクロ、リスト、ルールは、Falcoルールファイル内の順番を踏襲しています。Falcoルールの基本については、https://falco.org/jp/docs/rules/ を参照ください。

```
- required_engine_version: 2
```

共通マクロ、リスト - 1

Kubernetes Audit用のFalcoルール（ https://github.com/falcosecurity/falco/blob/master/rules/k8s_audit_rules.yaml）では、初めにFalcoルール全体で共通して利用するマクロとリストが定義されています。

always_true/always_false のように、k8 の監査イベントで動作します。

```
# always_true/always_false のように、k8 の監査イベントで動作します。
- macro: k8s_audit_always_true
  condition: (jevt.rawtime exists)

- macro: k8s_audit_never_true
  condition: (jevt.rawtime=0)
```

一般的には、レスポンスが完了してから監査イベントを検討することになります。

```
- list: k8s_audit_stages
  items: ["ResponseComplete"]
```

一般的に "system: "で始まるユーザを除外します。

```
- macro: non_system_user
  condition: (not ka.user.name startswith "system:")
```

Falcoルール - 1

1. **禁止されているK8sユーザー：Disallowed K8s User**
   許可されたユーザーセット以外のユーザーによるk8sの操作を検出します。

   このマクロは、以下のルールで使用される監査イベントのセットを選択します。

```
 - macro: kevt
   condition: (jevt.value[/stage] in (k8s_audit_stages))

 - macro: kevt_started
   condition: (jevt.value[/stage]=ResponseStarted)
```

   特定のユーザーにアクティビティを制限したい場合は、このリストを上書き/追加します。
   kopsによって作成されたユーザーが含まれています。

```
- list: vertical_pod_autoscaler_users
  items: ["vpa-recommender", "vpa-updater"]

- list: allowed_k8s_users
  items: [
    "minikube", "minikube-user", "kubelet", "kops", "admin", "kube", "kube-proxy", "kube-apiserver-healthcheck",
    "kubernetes-admin",
    vertical_pod_autoscaler_users,
    cluster-autoscaler,
    "system:addon-manager",
    "cloud-controller-manager"
    ]

- rule: Disallowed K8s User
  desc: Detect any k8s operation by users outside of an allowed set of users.
  condition: kevt and non_system_user and not ka.user.name in (allowed_k8s_users)
  output: K8s Operation performed by user not in allowed list of users (user=%ka.user.name
target=%ka.target.name/%ka.target.resource verb=%ka.verb uri=%ka.uri resp=%ka.response.code)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

共通マクロ、リスト - 2

ローカル/ユーザールールファイルでは、このマクロをオーバーライドして、環境で実行したいコンテナイメージを明示的に列挙することができます。このメインの falco ルールファイルでは、実行可能なすべてのコンテナを知る方法はありませんので、always_true マクロを使用することで、どのコンテナも許可されます。オーバーライドされたマクロでは、条件は次のようになります (ka.req.pod.containers.image.repository in (my-repo/my-image))

```
- macro: allowed_k8s_containers
  condition: (k8s_audit_always_true)

- macro: response_successful
  condition: (ka.response.code startswith 2)

- macro: kcreate
  condition: ka.verb=create

- macro: kmodify
  condition: (ka.verb in (create,update,patch))

- macro: kdelete
  condition: ka.verb=delete

- macro: pod
  condition: ka.target.resource=pods and not ka.target.subresource exists

- macro: pod_subresource
  condition: ka.target.resource=pods and ka.target.subresource exists

- macro: deployment
  condition: ka.target.resource=deployments

- macro: service
  condition: ka.target.resource=services

- macro: configmap
  condition: ka.target.resource=configmaps

- macro: namespace
  condition: ka.target.resource=namespaces

- macro: serviceaccount
  condition: ka.target.resource=serviceaccounts

- macro: clusterrole
  condition: ka.target.resource=clusterroles

- macro: clusterrolebinding
  condition: ka.target.resource=clusterrolebindings
```

```
- macro: role
  condition: ka.target.resource=roles

- macro: secret
  condition: ka.target.resource=secrets

- macro: health_endpoint
  condition: ka.uri=/healthz
```

Falcoルール - 2

2. **禁止されたポッドの作成：Create Disallowed Pod**
   許可されているイメージのリスト外のコンテナイメージでポッドを起動しようとしたことを検出します。

```
- rule: Create Disallowed Pod
  desc: >
    Detect an attempt to start a pod with a container image outside of a list of allowed images.
  condition: kevt and pod and kcreate and not allowed_k8s_containers
  output: Pod started with container not in allowed list (user=%ka.user.name pod=%ka.resp.name
ns=%ka.target.namespace images=%ka.req.pod.containers.image)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

3. **特権ポッドの作成：Create Privileged Pod**
   特権コンテナでポッドを起動しようとする試みを検出

```
- rule: Create Privileged Pod
  desc: >
    Detect an attempt to start a pod with a privileged container
  condition: kevt and pod and kcreate and ka.req.pod.containers.privileged intersects (true) and not
ka.req.pod.containers.image.repository in (falco_privileged_images)
  output: Pod started with privileged container (user=%ka.user.name pod=%ka.resp.name ns=%ka.target.namespace
images=%ka.req.pod.containers.image)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

4. **センシティブなマウントポッドを作成：Create Sensitive Mount Pod**
   センシティブなホストディレクトリ (/proc など) からのボリュームでポッドを起動しようとする試みを検出します。

```
- rule: Create Sensitive Mount Pod
  desc: >
    Detect an attempt to start a pod with a volume from a sensitive host directory (i.e. /proc).
    Exceptions are made for known trusted images.
  condition: kevt and pod and kcreate and sensitive_vol_mount and not ka.req.pod.containers.image.repository in
(falco_sensitive_mount_images)
  output: Pod started with sensitive mount (user=%ka.user.name pod=%ka.resp.name ns=%ka.target.namespace
images=%ka.req.pod.containers.image volumes=%jevt.value[/requestObject/spec/volumes])
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

5. **ホストネットワーク ポッドの作成：Create HostNetwork Pod**

   ホストネットワークを使用してポッドを起動しようとする試みを検出します。

   K8s CIS Benchmark 1.7.4への対応。

```
- rule: Create HostNetwork Pod
  desc: Detect an attempt to start a pod using the host network.
  condition: kevt and pod and kcreate and ka.req.pod.host_network intersects (true) and not
ka.req.pod.containers.image.repository in (falco_hostnetwork_images)
  output: Pod started using host network (user=%ka.user.name pod=%ka.resp.name ns=%ka.target.namespace
images=%ka.req.pod.containers.image)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

6. **NodePortサービスの作成：Create NodePort Service**

   NodePortサービスタイプでサービスを開始しようとしたことを検出します。

```
- macro: user_known_node_port_service
  condition: (k8s_audit_never_true)

- rule: Create NodePort Service
  desc: >
    Detect an attempt to start a service with a NodePort service type
  condition: kevt and service and kcreate and ka.req.service.type=NodePort and not user_known_node_port_service
  output: NodePort Service Created (user=%ka.user.name service=%ka.target.name ns=%ka.target.namespace
ports=%ka.req.service.ports)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

7. **プライベート認証情報を使用したConfigmapの作成/変更：Create/Modify Configmap With Private Credentials**

   プライベート認証情報(awsキー、パスワードなど)を含むconfigmapの作成/変更を検出

```
- macro: contains_private_credentials
  condition: >
    (ka.req.configmap.obj contains "aws_access_key_id" or
     ka.req.configmap.obj contains "aws-access-key-id" or
     ka.req.configmap.obj contains "aws_s3_access_key_id" or
     ka.req.configmap.obj contains "aws-s3-access-key-id" or
     ka.req.configmap.obj contains "password" or
     ka.req.configmap.obj contains "passphrase")
- rule: Create/Modify Configmap With Private Credentials
  desc: >
     Detect creating/modifying a configmap containing a private credential (aws key, password, etc.)
  condition: kevt and configmap and kmodify and contains_private_credentials
  output: K8s configmap with private credential (user=%ka.user.name verb=%ka.verb
configmap=%ka.req.configmap.name config=%ka.req.configmap.obj)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

8. **匿名リクエスト許可：Anonymous Request Allowed**

   匿名ユーザーが許可したリクエストを検出する

   K8s CIS Benchmark 1.1.1への対応。

```
- rule: Anonymous Request Allowed
  desc: >
    Detect any request made by the anonymous user that was allowed
  condition: kevt and ka.user.name=system:anonymous and ka.auth.decision="allow" and not health_endpoint
  output: Request by anonymous user allowed (user=%ka.user.name verb=%ka.verb uri=%ka.uri
reason=%ka.auth.reason))
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

9. **アタッチ/実行ポッド：Attach/Exec Pod**

   ポッドへのアタッチ/実行の試みを検出

   K8s CIS Benchmark, 1.1.12へおおよそ対応します。この場合、特権コンテナへの実行/アタッチの試みを通知します。

   理想的には、特権ポッドへのアタッチ/実行を検出するような、より厳しいルールを追加したいところですが、そのためには、k8sの監査イベントのエンジンがステートフルである必要があり、アタッチリクエストで指定されたコンテナが特権コンテナとして作成されたかどうかを知ることができます。今のところ、任意のポッドへのアタッチ/実行を検出する、それほど厳しくないルールを用意しています。

```
- macro: user_known_exec_pod_activities
```

```
  condition: (k8s_audit_never_true)

- rule: Attach/Exec Pod
  desc: >
    Detect any attempt to attach/exec to a pod
  condition: kevt_started and pod_subresource and kcreate and ka.target.subresource in (exec,attach) and not
user_known_exec_pod_activities
  output: Attach/Exec to pod (user=%ka.user.name pod=%ka.target.name ns=%ka.target.namespace
action=%ka.target.subresource command=%ka.uri.param[command])
  priority: NOTICE
  source: k8s_audit
  tags: [k8s]
```

10. エフェメラルコンテナの作成：EphemeralContainers Created

作成されたエフェメラルコンテナの作成を検出

```
- macro: user_known_pod_debug_activities
  condition: (k8s_audit_never_true)

# フィーチャーゲートEphemeralContainersが有効な場合にのみ動作します。
- rule: EphemeralContainers Created
  desc: >
    Detect any ephemeral container created
  condition: kevt and pod_subresource and kmodify and ka.target.subresource in (ephemeralcontainers) and not
user_known_pod_debug_activities
  output: Ephemeral container is created in pod (user=%ka.user.name pod=%ka.target.name ns=%ka.target.namespace
ephemeral_container_name=%jevt.value[/requestObject/ephemeralContainers/0/name]
ephemeral_container_image=%jevt.value[/requestObject/ephemeralContainers/0/image])
  priority: NOTICE
  source: k8s_audit
  tags: [k8s]
```

11. 禁止されたネームスペースの作成：Create Disallowed Namespace

既知のネームスペースのセットの外にネームスペースを作成しようとする試みを検出します。

ローカル/ユーザルールでは、このリストに追加して許可されるネームスペースを追加することができます。

```
- list: allowed_namespaces
  items: [kube-system, kube-public, default]

- rule: Create Disallowed Namespace
  desc: Detect any attempt to create a namespace outside of a set of known namespaces
  condition: kevt and namespace and kcreate and not ka.target.name in (allowed_namespaces)
  output: Disallowed namespace created (user=%ka.user.name ns=%ka.target.name)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

12. Kubeネームスペースにポッドが作成された：Pod Created in Kube Namespace

kube-system または kube-public ネームスペースでポッドを作成しようとする試みを検出します。

下位互換性のためにのみ定義されています。代わりに、より特定の user_allowed_kube_namespace_image_list を使用してください。

```
- list: user_trusted_image_list
  items: []

- list: user_allowed_kube_namespace_image_list
  items: [user_trusted_image_list]
```

下位互換性のためにのみ定義されています。代わりに、より具体的な allowed_kube_namespace_image_list を使用してください。

```
- list: k8s_image_list
  items: []

- list: allowed_kube_namespace_image_list
  items: [
    gcr.io/google-containers/prometheus-to-sd,
    gcr.io/projectcalico-org/node,
    gke.gcr.io/addon-resizer,
    gke.gcr.io/heapster,
    gke.gcr.io/gke-metadata-server,
    k8s.gcr.io/ip-masq-agent-amd64,
    k8s.gcr.io/kube-apiserver,
    gke.gcr.io/kube-proxy,
    gke.gcr.io/netd-amd64,
    k8s.gcr.io/addon-resizer
    k8s.gcr.io/prometheus-to-sd,
    k8s.gcr.io/k8s-dns-dnsmasq-nanny-amd64,
    k8s.gcr.io/k8s-dns-kube-dns-amd64,
    k8s.gcr.io/k8s-dns-sidecar-amd64,
    k8s.gcr.io/metrics-server-amd64,
    kope/kube-apiserver-healthcheck,
    k8s_image_list
    ]

- macro: allowed_kube_namespace_pods
  condition: (ka.req.pod.containers.image.repository in (user_allowed_kube_namespace_image_list) or
              ka.req.pod.containers.image.repository in (allowed_kube_namespace_image_list))
```

kube-system ネームスペースで作成された新しいポッドを検出します。

```
- rule: Pod Created in Kube Namespace
  desc: Detect any attempt to create a pod in the kube-system or kube-public namespaces
  condition: kevt and pod and kcreate and ka.target.namespace in (kube-system, kube-public) and not
allowed_kube_namespace_pods
  output: Pod created in kube namespace (user=%ka.user.name pod=%ka.resp.name ns=%ka.target.namespace
images=%ka.req.pod.containers.image)
```

```
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

13. Kube ネームスペースに作成されたサービスアカウント：Service Account Created in Kube Namespace

kube-system または kube-public ネームスペースにサービスアカウントを作成しようとする試みを検出します。

```
- list: user_known_sa_list
  items: []

- macro: trusted_sa
  condition: (ka.target.name in (user_known_sa_list))

# kube-system/kube-public ネームスペースでのサービスアカウントの作成を検出します。
- rule: Service Account Created in Kube Namespace
  desc: Detect any attempt to create a serviceaccount in the kube-system or kube-public namespaces
  condition: kevt and serviceaccount and kcreate and ka.target.namespace in (kube-system, kube-public) and
response_successful and not trusted_sa
  output: Service account created in kube namespace (user=%ka.user.name serviceaccount=%ka.target.name
ns=%ka.target.namespace)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

14. System ClusterRoleの変更/削除：System ClusterRole Modified/Deleted

システムで始まる ClusterRole/Role の変更/削除の試みを検出します。

"system: "で始まるClusterRoleへの変更/削除を検出します。"system:coredns" は通常の運用では変更が予想されるため除外されます。

```
- rule: System ClusterRole Modified/Deleted
  desc: Detect any attempt to modify/delete a ClusterRole/Role starting with system
  condition: kevt and (role or clusterrole) and (kmodify or kdelete) and (ka.target.name startswith "system:")
and
             not ka.target.name in (system:coredns, system:managed-certificate-controller)
  output: System ClusterRole/Role modified or deleted (user=%ka.user.name role=%ka.target.name
ns=%ka.target.namespace action=%ka.verb)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

15. cluster-admin Roleへのアタッチ：Attach to cluster-admin Role

ClusterRoleBinding を cluster-admin ユーザに作成しようとする試みを検出します。

ClusterRoleBinding を cluster-admin ユーザに作成しようとする試みを検出します (これを "センシティブ "なことを行う組み込みのクラスタロールに適用します)

```
- rule: Attach to cluster-admin Role
  desc: Detect any attempt to create a ClusterRoleBinding to the cluster-admin user
  condition: kevt and clusterrolebinding and kcreate and ka.req.binding.role=cluster-admin
  output: Cluster Role Binding to cluster-admin role (user=%ka.user.name subject=%ka.req.binding.subjects)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

16. ワイルドカードで作成されたClusterRole：ClusterRole With Wildcard Created

ワイルドカードリソースやverbsを使用して Role/ClusterRole を作成しようとする試みを検出します。

```
- rule: ClusterRole With Wildcard Created
  desc: Detect any attempt to create a Role/ClusterRole with wildcard resources or verbs
  condition: kevt and (role or clusterrole) and kcreate and (ka.req.role.rules.resources intersects ("*") or
ka.req.role.rules.verbs intersects ("*"))
  output: Created Role/ClusterRole with wildcard (user=%ka.user.name role=%ka.target.name
rules=%ka.req.role.rules)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

17. 書き込み権限を持つ ClusterRoleの作成：ClusterRole With Write Privileges Created

書き込み関連のアクションを実行できる Role/ClusterRole を作成しようとする試みを検出します。

```
- macro: writable_verbs
  condition: >
    (ka.req.role.rules.verbs intersects (create, update, patch, delete, deletecollection))
- rule: ClusterRole With Write Privileges Created
  desc: Detect any attempt to create a Role/ClusterRole that can perform write-related actions
  condition: kevt and (role or clusterrole) and kcreate and writable_verbs
  output: Created Role/ClusterRole with write privileges (user=%ka.user.name role=%ka.target.name
rules=%ka.req.role.rules)
  priority: NOTICE
  source: k8s_audit
  tags: [k8s]
```

18. Pod ExecでのClusterRole作成：ClusterRole With Pod Exec Created

ポッドに実行できる Role/ClusterRole を作成しようとする試みを検出します。

```
- rule: ClusterRole With Pod Exec Created
  desc: Detect any attempt to create a Role/ClusterRole that can exec to pods
```

```
  condition: kevt and (role or clusterrole) and kcreate and ka.req.role.rules.resources intersects ("pods/exec")
  output: Created Role/ClusterRole with pod exec privileges (user=%ka.user.name role=%ka.target.name
rules=%ka.req.role.rules)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

19. K8sデプロイメントの作成：K8s Deployment Created

デプロイメントを作成しようとする試みを検出します。

この点以下のルールは差別性が低く、一般的にクラスターの活動の流れを表しています。これらのイベントを無効にしたい場合は、以下のマクロを修正します。

```
- macro: consider_activity_events
  condition: (k8s_audit_always_true)

- macro: kactivity
  condition: (kevt and consider_activity_events)

- rule: K8s Deployment Created
  desc: Detect any attempt to create a deployment
  condition: (kactivity and kcreate and deployment and response_successful)
  output: K8s Deployment Created (user=%ka.user.name deployment=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

20. K8sデプロイメントの削除：K8s Deployment Deleted

デプロイメントを削除しようとする試みを検出します。

```
- rule: K8s Deployment Deleted
  desc: Detect any attempt to delete a deployment
  condition: (kactivity and kdelete and deployment and response_successful)
  output: K8s Deployment Deleted (user=%ka.user.name deployment=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

21. K8sサービスの作成：K8s Service Created

サービスを作成しようとする試みを検出します。

```
- rule: K8s Service Created
  desc: Detect any attempt to create a service
  condition: (kactivity and kcreate and service and response_successful)
```

```
  output: K8s Service Created (user=%ka.user.name service=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

22. K8sサービスの削除：K8s Service Deleted

サービスを削除しようとする試みを検出します。

```
- rule: K8s Service Deleted
  desc: Detect any attempt to delete a service
  condition: (kactivity and kdelete and service and response_successful)
  output: K8s Service Deleted (user=%ka.user.name service=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

23. K8s ConfigMapの作成：K8s ConfigMap Created

ConfigMapを作成しようとする試みを検出します。

```
- rule: K8s ConfigMap Created
  desc: Detect any attempt to create a configmap
  condition: (kactivity and kcreate and configmap and response_successful)
  output: K8s ConfigMap Created (user=%ka.user.name configmap=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

24. K8s ConfigMapの削除：K8s ConfigMap Deleted

ConfigMapを削除しようとする試みを検出します。

```
- rule: K8s ConfigMap Deleted
  desc: Detect any attempt to delete a configmap
  condition: (kactivity and kdelete and configmap and response_successful)
  output: K8s ConfigMap Deleted (user=%ka.user.name configmap=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

25. K8sネームスペースの作成：K8s Namespace Created

ネームスペースを作成しようとする試みを検出します。

```
- rule: K8s Namespace Created
  desc: Detect any attempt to create a namespace
  condition: (kactivity and kcreate and namespace and response_successful)
  output: K8s Namespace Created (user=%ka.user.name namespace=%ka.target.name resp=%ka.response.code
decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

26. K8sネームスペースの削除：K8s Namespace Deleted

ネームスペースを削除しようとする試みを検出します。

```
- rule: K8s Namespace Deleted
  desc: Detect any attempt to delete a namespace
  condition: (kactivity and non_system_user and kdelete and namespace and response_successful)
  output: K8s Namespace Deleted (user=%ka.user.name namespace=%ka.target.name resp=%ka.response.code
decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

27. K8sサービスアカウントの作成：K8s Serviceaccount Created

サービスアカウントを作成しようとする試みを検出します。

```
- rule: K8s Serviceaccount Created
  desc: Detect any attempt to create a service account
  condition: (kactivity and kcreate and serviceaccount and response_successful)
  output: K8s Serviceaccount Created (user=%ka.user.name user=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

28. K8sサービスアカウントの削除：K8s Serviceaccount Deleted

サービスアカウントを削除しようとする試みを検出します。

```
- rule: K8s Serviceaccount Deleted
  desc: Detect any attempt to delete a service account
  condition: (kactivity and kdelete and serviceaccount and response_successful)
  output: K8s Serviceaccount Deleted (user=%ka.user.name user=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

29. K8sロール/クラスターロールの作成：K8s Role/Clusterrole Created

ロール/クラスターロールを作成しようとする試みを検出します。

```
- rule: K8s Role/Clusterrole Created
  desc: Detect any attempt to create a cluster role/role
  condition: (kactivity and kcreate and (clusterrole or role) and response_successful)
  output: K8s Cluster Role Created (user=%ka.user.name role=%ka.target.name rules=%ka.req.role.rules
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

30. K8sロール/クラスターロールの削除：K8s Role/Clusterrole Deleted

ロール/クラスターロールを削除しようとする試みを検出します。

```
- rule: K8s Role/Clusterrole Deleted
  desc: Detect any attempt to delete a cluster role/role
  condition: (kactivity and kdelete and (clusterrole or role) and response_successful)
  output: K8s Cluster Role Deleted (user=%ka.user.name role=%ka.target.name resp=%ka.response.code
decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

31. K8sクラスターロールバインディングの作成：K8s Role/Clusterrolebinding Created

クラスターロールバインディングを作成しようとする試みを検出します。

```
- rule: K8s Role/Clusterrolebinding Created
  desc: Detect any attempt to create a clusterrolebinding
  condition: (kactivity and kcreate and clusterrolebinding and response_successful)
  output: K8s Cluster Role Binding Created (user=%ka.user.name binding=%ka.target.name
subjects=%ka.req.binding.subjects role=%ka.req.binding.role resp=%ka.response.code decision=%ka.auth.decision
reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

32. K8sクラスターロールバインディングの削除：K8s Role/Clusterrolebinding Deleted

クラスターロールバインディングを削除しようとする試みを検出します。

```
- rule: K8s Role/Clusterrolebinding Deleted
  desc: Detect any attempt to delete a clusterrolebinding
  condition: (kactivity and kdelete and clusterrolebinding and response_successful)
```

```
  output: K8s Cluster Role Binding Deleted (user=%ka.user.name binding=%ka.target.name resp=%ka.response.code
decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

33. K8sシークレットの作成：K8s Secret Created

シークレットを作成しようとする試みを検出します。サービスアカウントトークンは除外されます。

```
- rule: K8s Secret Created
  desc: Detect any attempt to create a secret. Service account tokens are excluded.
  condition: (kactivity and kcreate and secret and ka.target.namespace!=kube-system and non_system_user and
response_successful)
  output: K8s Secret Created (user=%ka.user.name secret=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

34. K8sシークレットの削除：K8s Secret Deleted

シークレットを削除しようとする試みを検出します。サービスアカウントトークンは除外されます。

```
- rule: K8s Secret Deleted
  desc: Detect any attempt to delete a secret Service account tokens are excluded.
  condition: (kactivity and kdelete and secret and ka.target.namespace!=kube-system and non_system_user and
response_successful)
  output: K8s Secret Deleted (user=%ka.user.name secret=%ka.target.name ns=%ka.target.namespace
resp=%ka.response.code decision=%ka.auth.decision reason=%ka.auth.reason)
  priority: INFO
  source: k8s_audit
  tags: [k8s]
```

35. 全てのK8s Auditイベント：All K8s Audit Events

全てのK8s Auditイベントを検出

このルールは一般的にすべてのイベントにマッチし、その結果、デフォルトでは無効になっています。これらのイベントを有効にしたい場合は、以下のマクロを修正します。

condition: (jevt.rawtime exists)

```
- macro: consider_all_events
  condition: (k8s_audit_never_true)
```

```
- macro: kall
  condition: (kevt and consider_all_events)

- rule: All K8s Audit Events
  desc: Match all K8s Audit Events
  condition: kall
  output: K8s Audit Event received (user=%ka.user.name verb=%ka.verb uri=%ka.uri obj=%jevt.obj)
  priority: DEBUG
  source: k8s_audit
  tags: [k8s]
```

36. フルK8s管理者アクセス：Full K8s Administrative Access

フルアクセスの管理者と思われるユーザ名でk8sの操作を検出します。

このマクロは以下のルールを無効にし、有効にするには k8s_audit_never_true に変更します。

```
- macro: allowed_full_admin_users
  condition: (k8s_audit_always_true)
```

このリストには、いくつかの K8 のインストールにおける管理者のデフォルトユーザ名が含まれています。

```
- list: full_admin_k8s_users
  items: ["admin", "kubernetes-admin",  "kubernetes-admin@kubernetes", "kubernetes-admin@cluster.local",
"minikube-user"]
```

このルールは、クラスタ作成時のデフォルト管理者のリストに含まれるユーザ名によって引き起こされた操作を検出します。これは権限の設定が広すぎることを示しています。一般的なka.*イベントではユーザのロールを確認できないので、管理者ではないかもしれません。
full_admin_k8s_usersリストをカスタマイズして、必要に応じて有効化してください。

テストの方法：
デフォルトのクラスタユーザで接続された任意の kubectl コマンドを実行します：
kubectl create namespace rule-test

```
- rule: Full K8s Administrative Access
  desc: Detect any k8s operation by a user name that may be an administrator with full access.
  condition: >
    kevt
    and non_system_user
    and ka.user.name in (full_admin_k8s_users)
    and not allowed_full_admin_users
  output: K8s Operation performed by full admin user (user=%ka.user.name
target=%ka.target.name/%ka.target.resource verb=%ka.verb uri=%ka.uri resp=%ka.response.code)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```

37. TLS証明書のないIngressオブジェクトの作成：Ingress Object without TLS Certificate Created

TLS 認証なしでIngressを作成しようとする試みを検出します。

```
- macro: ingress
  condition: ka.target.resource=ingresses

- macro: ingress_tls
  condition: (jevt.value[/requestObject/spec/tls] exists)
```

テストの方法：

コンテンツを含むingress.yamlファイルを作成します。

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
 name: test-ingress
 annotations:
  nginx.ingress.kubernetes.io/rewrite-target: /
spec:
 rules:
 - http:
   paths:
   - path: /testpath
    backend:
     serviceName: test
     servicePort: 80
```

kubectl apply -f ingress.yamlを実行

```
- rule: Ingress Object without TLS Certificate Created
  desc: Detect any attempt to create an ingress without TLS certification.
  condition: >
    (kactivity and kcreate and ingress and response_successful and not ingress_tls)
  output: >
    K8s Ingress Without TLS Cert Created (user=%ka.user.name ingress=%ka.target.name
    namespace=%ka.target.namespace)
  source: k8s_audit
  priority: WARNING
  tags: [k8s, network]
```

38. 信頼されていないノードが正常にクラスタに参加：Untrusted Node Successfully Joined the Cluster

許可されたノードのリストの外にあるノードが正常にクラスターに参加したことを検出します。

テストの方法：

Kopsを使用してFalco監視クラスタを作成します。

kops edit ig nodes

kops apply --yes

```
- rule: Untrusted Node Successfully Joined the Cluster
  desc: >
    Detect a node successfully joined the cluster outside of the list of allowed nodes.
  condition: >
    kevt and node
    and kcreate
    and response_successful
    and not allow_all_k8s_nodes
    and not ka.target.name in (allowed_k8s_nodes)
  output: Node not in allowed list successfully joined the cluster (user=%ka.user.name node=%ka.target.name)
  priority: ERROR
  source: k8s_audit
  tags: [k8s]
```

39. 信頼されていないノードがクラスタに参加しようとして失敗：Untrusted Node Unsuccessfully Tried to Join the Cluster

許可されているノードのリストにないノードのクラスタへの参加に失敗したことを検出します。

```
- rule: Untrusted Node Unsuccessfully Tried to Join the Cluster
  desc: >
    Detect an unsuccessful attempt to join the cluster for a node not in the list of allowed nodes.
  condition: >
    kevt and node
    and kcreate
    and not response_successful
    and not allow_all_k8s_nodes
    and not ka.target.name in (allowed_k8s_nodes)
  output: Node not in allowed list tried unsuccessfully to join the cluster  (user=%ka.user.name
node=%ka.target.name reason=%ka.response.reason)
  priority: WARNING
  source: k8s_audit
  tags: [k8s]
```