|  |  |
| --- | --- |
| **DOCUMENT RULES:** | |
| **Task Number / Name:** | **Killer Shell CKS** |
| **Task name & column name should be written:** | **Bold (CTRL+B)** |
| **Commands should be written in the after # sign:** | *Italic (CTRL+I) #hostname* |
| **Output photo should be cropped or compressed:**  **Photo could be more than one:**  **If you need extra lines, add the line next after it:** | ***Description photo should be with title bar (CTRL + I + B)*** |
| **All other text should be written:** | Standard |
| **Font name and text size:** | Calibri and 9 |
| **Group name:** | Dev\_ops\_ |
| **Student name and surname:** | Murad Abbaszade |
| **E-mail:** | [muradabbaszade6@gmail.com](mailto:muradabbaszade6@gmail.com) |
| **WhatsApp number:** | **+994703664205** |

|  |  |
| --- | --- |
| 1.Vim setup |  |
| nano ~/.vimrc |  |
| 2.Apiserver Crash |  |
| cp ~/kube-apiserver.yaml.ori /etc/kubernetes/manifests/kube-apiserver.yaml  nano /etc/kubernetes/manifests/kube-apiserver.yaml |  |
| watch crictl ps  k -n kube-system get pod |  |
| cp ~/kube-apiserver.yaml.ori /etc/kubernetes/manifests/kube-apiserver.yaml |  |
| cp /etc/kubernetes/manifests/kube-apiserver.yaml ~/kube-apiserver.yaml.ori  nano /etc/kubernetes/manifests/kube-apiserver.yaml |  |
| watch crictl ps  k -n kube-system get pod |  |
| cp ~/kube-apiserver.yaml.ori /etc/kubernetes/manifests/kube-apiserver.yaml |  |
| Change the Apiserver manifest and add invalid YAML, something like this:  apiVersionTHIS IS VERY ::::: WRONG v1  kind: Pod  metadata:  nano /etc/kubernetes/manifests/kube-apiserver.yaml |  |
| watch crictl ps  k -n kube-system get pod  cp ~/kube-apiserver.yaml.ori /etc/kubernetes/manifests/kube-apiserver.yaml |  |
| 4.Apiserver Misconfigured |  |
| cat /var/log/syslog | grep kube-apiserver |  |
| 5.Apiserver Node restriction |  |
| ssh node01 |  |
| export KUBECONFIG=/etc/kubernetes/kubelet.conf |  |
| k label node controlplane killercoda/one=123 |  |
| k label node node01 node-restriction.kubernetes.io/one=123 |  |
| watch crictl ps |  |
| We need to enable the NodeRestriction in the Apiserver manifest | spec:  containers:  - command:  - kube-apiserver  - --advertise-address=172.30.1.2  - --allow-privileged=true  - --authorization-mode=Node,RBAC  - --client-ca-file=/etc/kubernetes/pki/ca.crt  **- --enable-admission-plugins=NodeRestriction**  - --enable-bootstrap-token-auth=true |
| ssh node01 |  |
| export KUBECONFIG=/etc/kubernetes/kubelet.conf  k label node controlplane killercoda/two=123 # restricted  k label node node01 node-restriction.kubernetes.io/two=123 # restricted  k label node node01 test/two=123 # works |  |
| 6.Api armor |  |
| The /root/profiles.txt on node controlplane should look like |  |
| ssh node01 |  |
| apparmor\_status |  |
| And the annotation needs to reference the correct container name httpd . |  |
| 7.Auditing Enable Audit Logging |  |
| Create the directory | mkdir /etc/kubernetes/audit-logs |
| Edit the /etc/kubernetes/manifests/kube-apiserver.yaml  volumes:  - name: audit-policy  hostPath:  path: /etc/kubernetes/audit-policy/policy.yaml  type: File  - name: audit-logs  hostPath:  path: /etc/kubernetes/audit-logs  type: DirectoryOrCreate |  |
| watch crictl ps |  |
| stat /etc/kubernetes/audit-logs/audit.log  cat /etc/kubernetes/manifests/kube-apiserver.yaml | grep audit-policy-file=/etc/kubernetes/audit-policy/policy.yaml  cat /etc/kubernetes/manifests/kube-apiserver.yaml | grep audit-log-path=/etc/kubernetes/audit-logs/audit.log  cat /etc/kubernetes/manifests/kube-apiserver.yaml | grep audit-log-maxsize=7  cat /etc/kubernetes/manifests/kube-apiserver.yaml | grep audit-log-maxbackup=2 |  |
| 8. Certificate signing request sign manually |  |
| openssl genrsa -out 60099.key 2048 |  |
| openssl req -new -key 60099.key -out 60099.csr |  |
| openssl x509 -req -in 60099.csr -CA /etc/kubernetes/pki/ca.crt -CAkey /etc/kubernetes/pki/ca.key -CAcreateserial -out 60099.crt -days 500 |  |
| k config set-credentials 60099@internal.users --client-key=60099.key --client-certificate=60099.crt |  |
| k config set-context 60099@internal.users --cluster=kubernetes --user=60099@internal.users |  |
| k config get-contexts |  |
| k config use-context 60099@internal.users |  |
| 9. Certificate signing requests-sign via APi |  |
| openssl genrsa -out 60099.key 2048 |  |
| openssl req -new -key 60099.key -out 60099.csr |  |
| Convert the CSR file into base64  cat 60099.csr | base64 -w 0 |  |
| Nano csr.yaml  apiVersion: certificates.k8s.io/v1  kind: CertificateSigningRequest  metadata:  name: 60099@internal.users # ADD  spec:  groups:  - system:authenticated  request: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSBSRVFV... # ADD  signerName: kubernetes.io/kube-apiserver-client  usages:  - client auth |  |
| k -f csr.yaml create  k get csr  k certificate approve 60099@internal.users  k get csr # approved  k get csr 60099@internal.users ojsonpath="{.status.certificate}" | base64 -d > 60099.crt |  |
| 10.CIS Benchmarks fix Controlplane |  |
| kube-bench run --targets master |  |
| kube-bench run --targets master --check 1.2.20 |  |
| ...  containers:  - command:  - kube-apiserver  - --profiling=false  ...  image: k8s.gcr.io/kube-apiserver:v1.22.2 |  |
| watch crictl ps |  |
| kube-bench run --targets master |  |
| kube-bench run --targets master --check 1.3.2 |  |
| Nano /etc/kubernetes/manifests/kube-controller-manager.yaml  ...  containers:  - command:  - kube-controller-manager  - --profiling=false  ...  image: k8s.gcr.io/kube-controller-manager:v1.22.2  ... |  |
| watch crictl ps |  |
| kube-bench run --targets master |  |
| kube-bench run --targets master --check 1.1.19  chgrp root /etc/kubernetes/pki/ |  |
| 11.Container hardening |  |
| Nano Dockerfile  FROM ubuntu**:20.04**  RUN apt-get update  RUN apt-get -y install curl  ENV URL https://google.com/this-will-fail?secret-token=  CMD ["sh", "-c", "curl --head $URL$TOKEN"] |  |
| Docker build . |  |
| 12.Container image footprint user |  |
| cd /opt/ks/  docker build -t base-image . |  |
| docker run --name c1 -d base-image |  |
| docker exec c1 ps |  |
| docker rm c1 --force |  |
| FROM alpine:3.12.3  RUN adduser -D -g '' appuser  USER appuser  CMD sh -c 'sleep 1d' |  |
| cd /opt/ks/  docker build -t base-image . |  |
| docker run --name c2 -d base-image  docker exec c2 ps |  |
| 13.Container Namespaces Docker |  |
| docker run --name app1 -d nginx:alpine sleep infinity |  |
| docker exec app1 ps aux |  |
| docker run --name app2 --pid=container:app1 -d nginx:alpine sleep infinity |  |
| docker exec app1 ps aux  docker exec app2 ps aux |  |
| 14.Container namespaces podman |  |
| podman run --name app1 -d nginx:alpine sleep infinity |  |
| podman exec app1 ps aux |  |
| podman run --name app2 --pid=container:app1 -d nginx:alpine sleep infinity |  |
| podman exec app1 ps aux  podman exec app2 ps aux |  |
| 15.Falco Change role |  |
| k run pod --image=nginx:alpine |  |
| k exec -it pod -- sh  cat /var/log/syslog | grep falco | grep shell |  |
| cd /etc/falco/ |  |
| cp falco\_rules.yaml falco\_rules.local.yaml |  |
| vim falco\_rules.local.yaml |  |
| service falco restart  k exec -it pod – sh  cat /var/log/syslog | grep falco | grep shell |  |
| 16.ImagePolicy Webhook Setup |  |
| Nano /etc/kubernetes/policywebhook/admission\_config.json |  |
| Nano  /etc/kubernetes/policywebhook/kubeconf |  |
| watch crictl ps |  |
| k run pod --image=nginx |  |
| 17.Image use digest |  |
| k run crazy-pod --image=nginx@sha256:eb05700fe7baa6890b74278e39b66b2ed1326831f9ec3ed4bdc6361a4ac2f333 |  |
| k get deploy --show-labels |  |
| k get pod -l app=crazy-deployment -oyaml | grep imageID |  |
| k edit deploy crazy-deployment # image: httpd@sha256:c7b8040505e2e63eafc82d37148b687ff488bf6d25fc24c8bf01d71f5b457531 |  |
| k get pod -l app=crazy-deployment -oyaml | grep image: |  |
| 18.Image vulnerability scanning  trivy |  |
| k -n applications get pod -oyaml | grep image: |  |
| trivy image nginx:1.19.1-alpine-perl | grep CVE-2021-28831 |  |
| trivy image nginx:1.19.1-alpine-perl | grep CVE-2016-9841  trivy image nginx:1.20.2-alpine | grep CVE-2021-28831  trivy image nginx:1.20.2-alpine | grep CVE-2016-9841 |  |
| k -n applications scale deploy web1 --replicas 0 |  |
| k -n infra get pod -oyaml | grep image: |  |
| trivy image httpd:2.4.39-alpine | grep CVE-2021-28831  trivy image httpd:2.4.39-alpine | grep CVE-2016-9841 |  |
| k -n infra scale deploy inf-hjk --replicas 0 |  |
| 19.Immutability ReadOnly filesystem |  |
| k -n sun run pod-ro --image=busybox:1.32.0 -oyaml --dry-run=client --command -- sh -c 'sleep 1d' > pod.yaml |  |
| Nano pod.yaml  apiVersion: v1  kind: Pod  metadata:  labels:  run: pod-ro  name: pod-ro  namespace: sun  spec:  containers:  - command:  - sh  - -c  - sleep 1d  image: busybox:1.32.0  name: pod-ro  securityContext:  readOnlyRootFilesystem: true  dnsPolicy: ClusterFirst  restartPolicy: Always |  |
| Kubectl create –f pod.yaml |  |
| 20.Ingress Create |  |
| k -n world expose deploy europe --port 80 |  |
| k -n world expose deploy asia --port 80 |  |
| nano world.yaml |  |
| Kubectl create –f world.yaml |  |
| 21.Ingress Secure |  |
| curl http://world.universe.mine:30080/europe |  |
| openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout cert.key -out cert.crt -subj "/CN=world.universe.mine/O=world.universe.mine" |  |
| kubectl -n world create secret tls ingress-tls --key cert.key --cert cert.crt |  |
| k -n world edit ing world |  |
| curl -m1 -kvI https://world.universe.mine:30443/europe 2>&1 | grep subject | grep world.universe.mine |  |
| 22.Network policy create default deny |  |
| Create the NP:  apiVersion: networking.k8s.io/v1  kind: NetworkPolicy  metadata:  name: deny-out  namespace: app  spec:  podSelector: {}  policyTypes:  - Egress  egress:  - ports:  - port: 53  protocol: TCP  - port: 53  protocol: UDP |  |
| Kubectl create –f np.yaml |  |
| k -n app exec app1-0 -- curl [www.google.com](http://www.google.com)  k -n app exec microservice1-0 -- curl www.google.com |  |
| k -n app exec app1-0 -- curl microservice1  k -n app exec microservice1-0 -- curl app1 |  |
| k -n default run nginx --image=nginx:1.21.5-alpine --restart=Never -i --rm -- curl app1.app.svc.cluster.local |  |
| k -n default run nginx --image=nginx:1.21.5-alpine --restart=Never -i --rm -- curl microservice1.app.svc.cluster.local |  |
| 23.Network policy metadata protection |  |
| Create the NP where we allow traffic to all addresses, except the evil one    apiVersion: networking.k8s.io/v1  kind: NetworkPolicy  metadata:  name: metadata-server  namespace: default  spec:  podSelector:  matchLabels:  trust: nope  policyTypes:  - Egress  egress:  - to:  - ipBlock:  cidr: 0.0.0.0/0  except:  - 1.1.1.1/32 |  |
| Kubectl create –f np.yaml |  |
| 24.Network Policy Namespace selector |  |
| Create the first NP:    apiVersion: networking.k8s.io/v1  kind: NetworkPolicy  metadata:  name: np  namespace: space1  spec:  podSelector: {}  policyTypes:  - Egress  egress:  - to:  - namespaceSelector:  matchLabels:  kubernetes.io/metadata.name: space2  - ports:  - port: 53  protocol: TCP  - port: 53  protocol: UDP |  |
| Kubectl create –f np.yaml |  |
| Create the second NP:    apiVersion: networking.k8s.io/v1  kind: NetworkPolicy  metadata:  name: np  namespace: space2  spec:  podSelector: {}  policyTypes:  - Ingress  ingress:  - from:  - namespaceSelector:  matchLabels:  kubernetes.io/metadata.name: space1 |  |
| Kubectl create –f np2.yaml |  |
| 25. |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |