



Securing Secrets

Threat Model and AccuKnox Protection



Certified & Accredited by



As Featured In



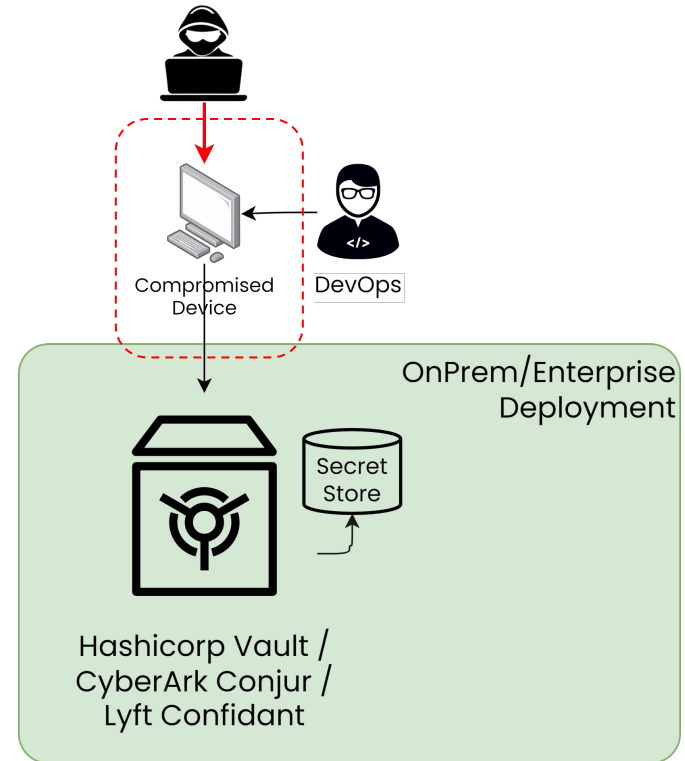
Available On



- **Start by looking at the Threat model for Secrets Management**
- **Identify attack possibilities**
 - Client side attack
 - Secrets store side attack possibilities
- **Understand how AccuKnox can help secure**
 - Protecting secrets stored in env vars, volume mount points
 - Least permissive granular system access

- **User-Access Threat Model**

Compromise of a user's computer can give the attacker access to all actions achievable by the authenticated user.



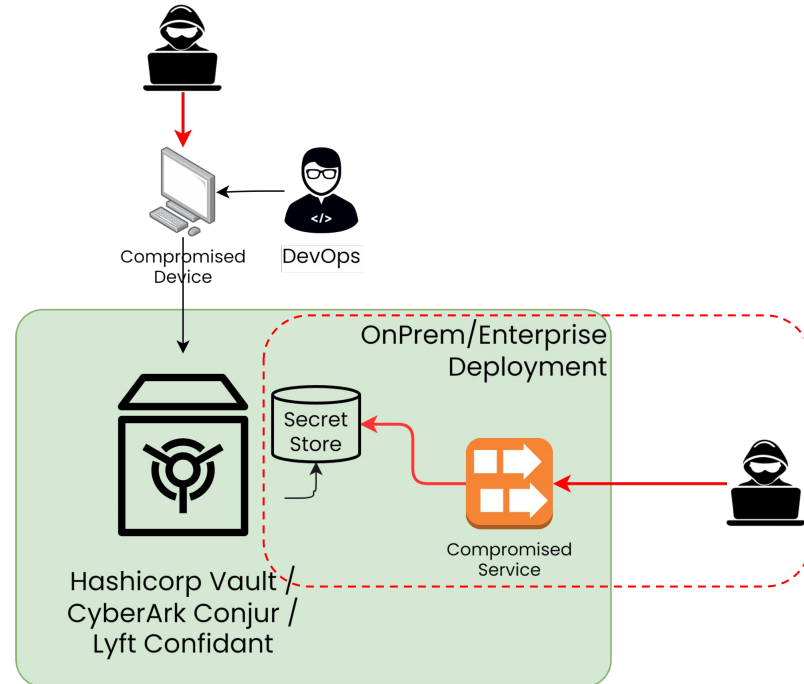
- **Server Threat Model**

An attacker who compromises the server has full control of the storage and can corrupt or delete secrets, encrypt secrets (ransomware), manipulate server logs.

Organizations deploying onprem secrets management have real risk of Ransomware attack.

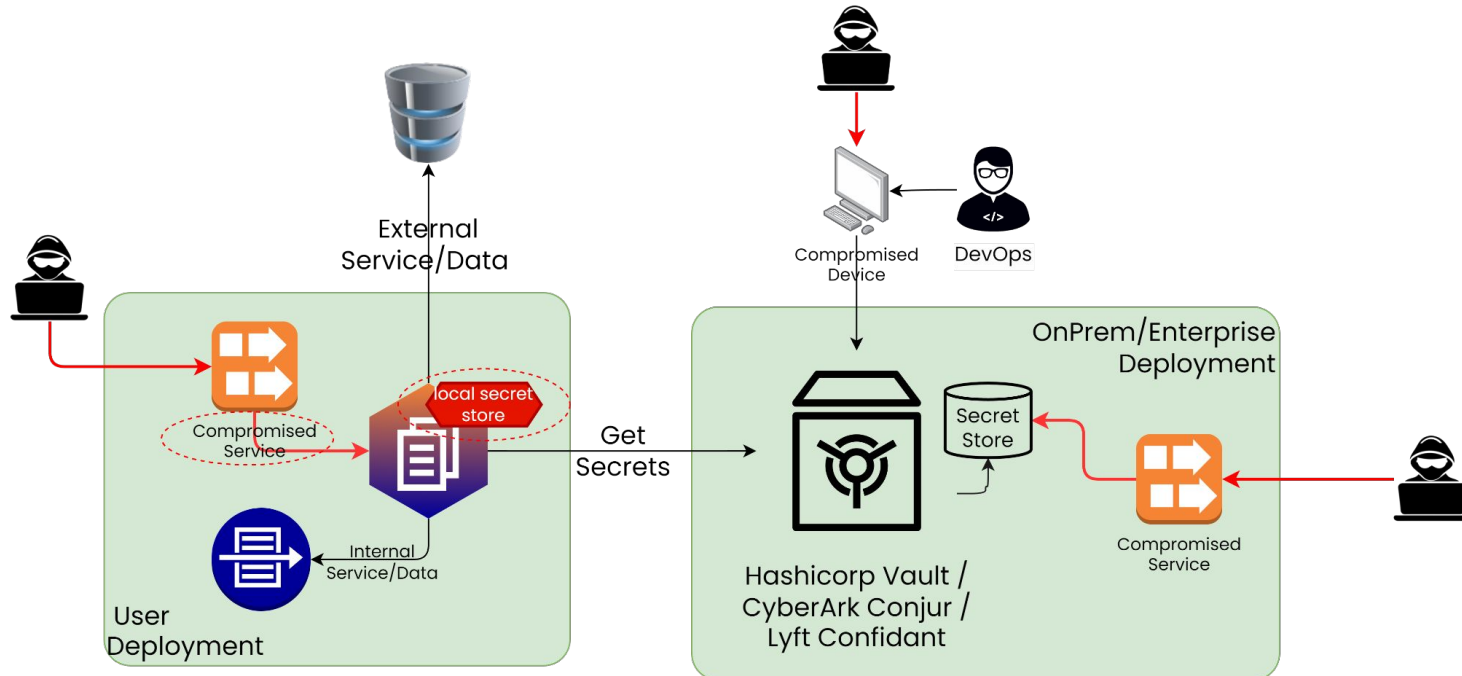
Implement a zero trust architecture to prevent unauthorized access to data and services. Make access control enforcement as granular as possible. ZTA assumes a network is compromised and provides a collection of concepts and ideas designed to minimize uncertainty in enforcing accurate, least privilege per request access decisions in information systems and services.

- <https://www.cisa.gov/stopransomware/ransomware-guide>

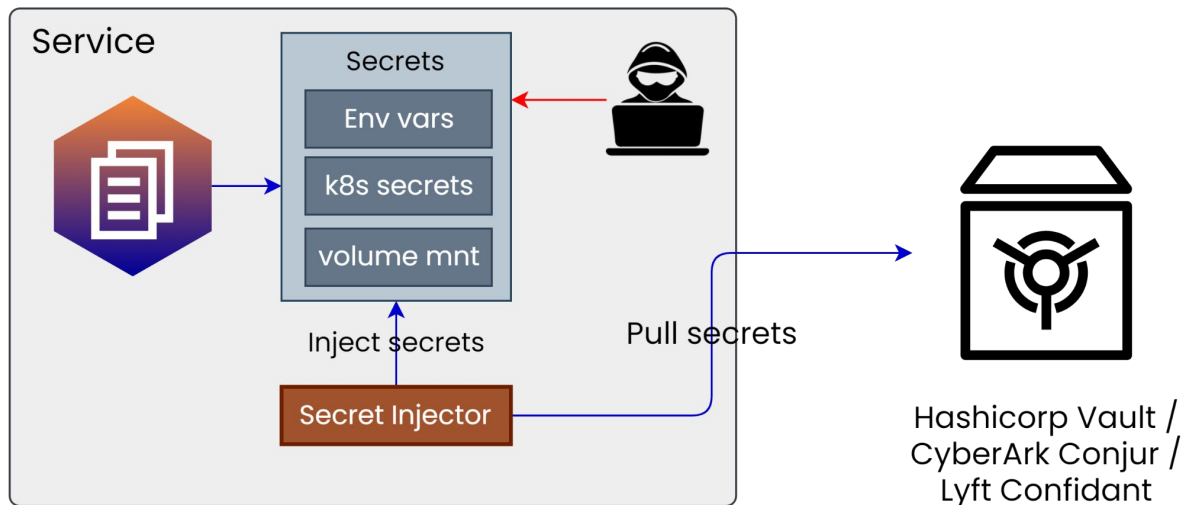


Threat Model for Secret Management

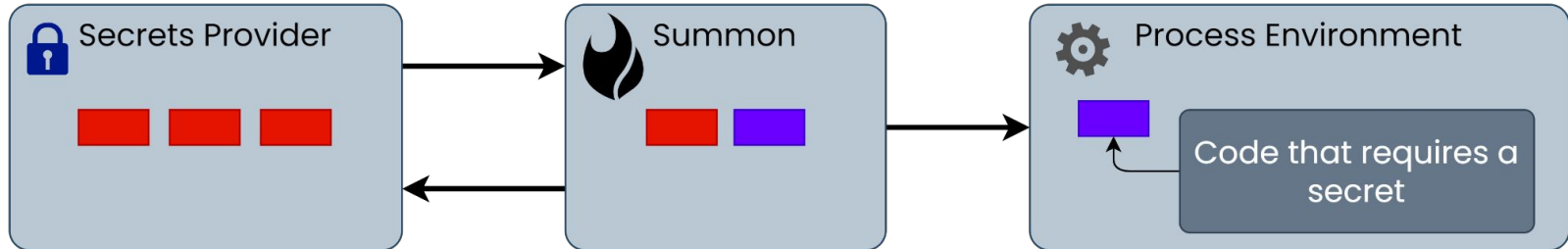
- **Client Threat Model: Get Secrets and store locally**
 - Storing the secrets on the filesystem
 - Storing the secrets as environment variables using injectors



- Popular Methods for Injecting secrets as:
 - Environment variables
 - K8s secrets
 - Volume mount points



- **CyberArk Summon**
 - Inject secrets from Conjur/KMS/Keepass into services as environment vars.
- **Hashicorp Vault Injector does similar handling for k8s clusters.**



How does AccuKnox tooling help secure?

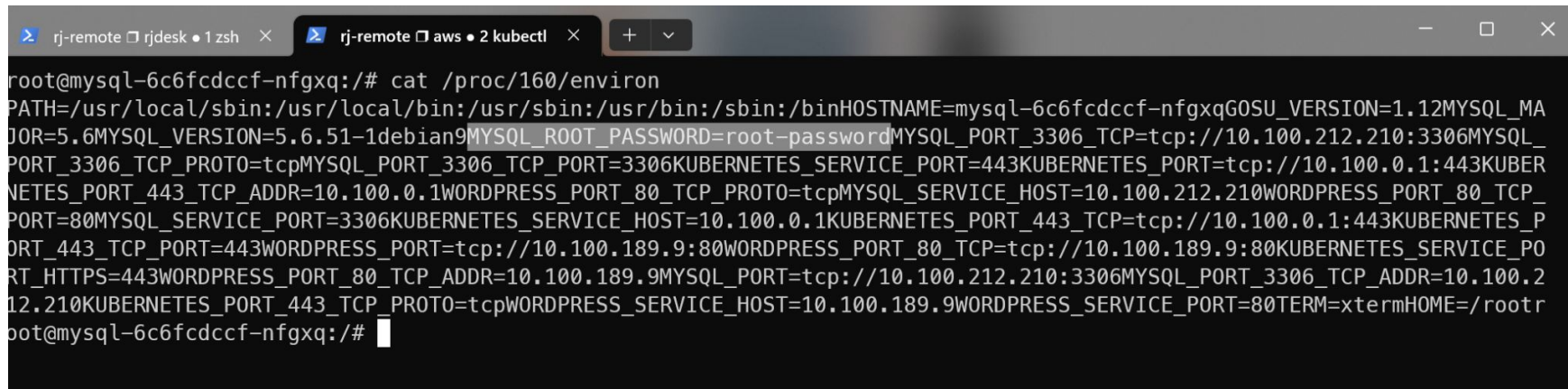
- External process can access other process env vars through *proc fs*
 - `/proc/{pid}/environ`
- KubeArmor can restrict access to `/proc/{pid}/environ` to only the target process identified by *process-name/{pid}*.

<https://attack.mitre.org/techniques/T1083/>

G0139

TeamTNT

TeamTNT has used a script that checks `/proc/*/environ` for environment variables related to AWS.^[301]



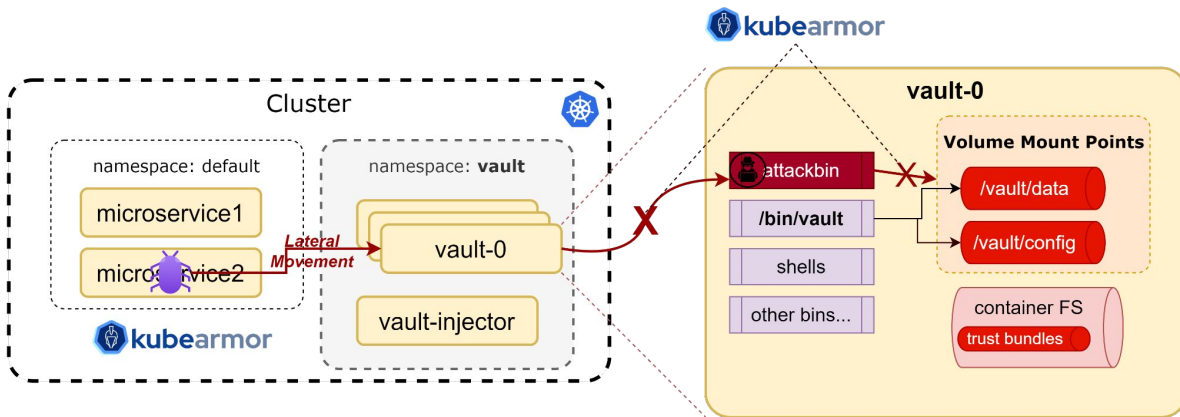
```
root@mysql-6c6fcdccf-nfgxq:/# cat /proc/160/environ
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/binHOSTNAME=mysql-6c6fcdccf-nfgxqGOSU_VERSION=1.12MYSQL_MAJOR=5.6MYSQL_VERSION=5.6.51-1debian9MYSQL_ROOT_PASSWORD=root-passwordMYSQL_PORT_3306_TCP=tcp://10.100.212.210:3306MYSQL_PORT_3306_TCP_PROTO=tcpMYSQL_PORT_3306_TCP_PORT=3306KUBERNETES_SERVICE_PORT=443KUBERNETES_PORT=tcp://10.100.0.1:443KUBERNETES_PORT_443_TCP_ADDR=10.100.0.1WORDPRESS_PORT_80_TCP_PROTO=tcpMYSQL_SERVICE_HOST=10.100.212.210WORDPRESS_PORT_80_TCP_PORT=80MYSQL_SERVICE_PORT=3306KUBERNETES_SERVICE_HOST=10.100.0.1KUBERNETES_PORT_443_TCP=tcp://10.100.0.1:443KUBERNETES_PORT_443_TCP_PORT=443WORDPRESS_PORT=tcp://10.100.189.9:80WORDPRESS_PORT_80_TCP=tcp://10.100.189.9:80KUBERNETES_SERVICE_PORT_HTTPS=443WORDPRESS_PORT_80_TCP_ADDR=10.100.189.9MYSQL_PORT=tcp://10.100.212.210:3306MYSQL_PORT_3306_TCP_ADDR=10.100.212.210KUBERNETES_PORT_443_TCP_PROTO=tcpWORDPRESS_SERVICE_HOST=10.100.189.9WORDPRESS_SERVICE_PORT=80TERM=xtermHOME=/root
root@mysql-6c6fcdccf-nfgxq:/#
```

- **Protecting access to environment vars**
 - Only allow the owning process access to its env vars
- **Do not allow unknown process execution**
 - Allow specific bins, deny unknown bins

```
apiVersion: security.kubearmor.com/v1
kind: KubeArmorPolicy
metadata:
  name: protect-env-vars
spec:
  selector:
    matchLabels:
      app: microservice
  file:
    matchDirectories:
      - dir: /proc/PID/environ
        fromSource:
          - path: /home/app/app-service
  preset-rule:
    - env-var-protect
    - fileless-process-exec
  process:
    matchPaths:
      - path: /home/app/app-service
  action:
    Allow
```

- **With KubeArmor one can protect any sensitive asset**
 - It could be k8s secrets mounted in the pod
 - Or any other volume mount point
- **Enforce constraints such as,**
 - Allow access to certain paths only for certain process(es)
 - Allow readOnly access to all but write access to certain process(es) only

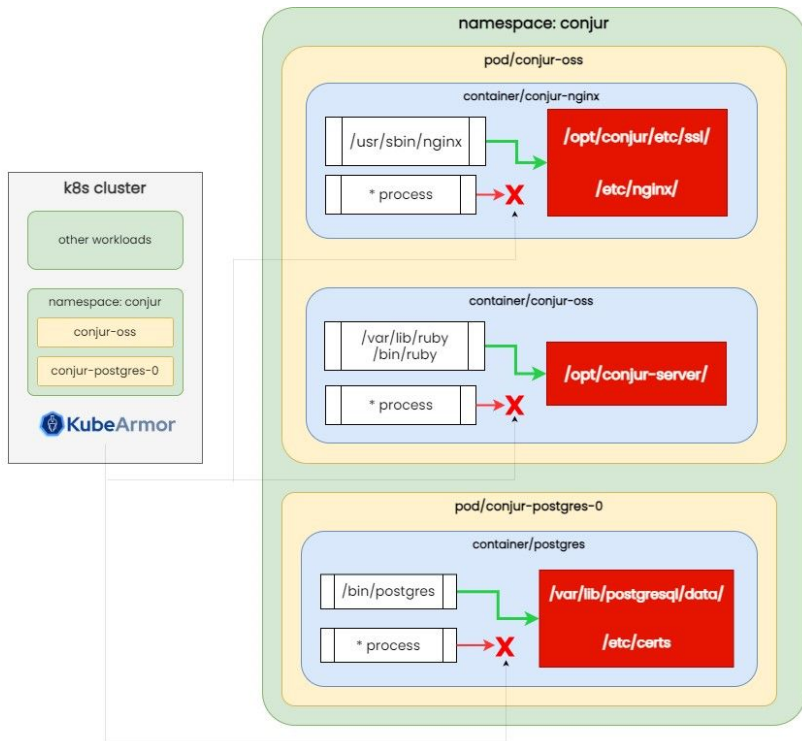
- Protecting onprem Vault
Real risk of Ransomware attacks



- Implement Least permissive access to sensitive assets

```
apiVersion: security.kubearmor.com/v1
kind: KubeArmorPolicy
metadata:
  name: ksp-vault-protect
  namespace: default
spec:
  severity: 7
  selector:
    matchLabels:
      app.kubernetes.io/name: vault
      component: server
  file:
    matchDirectories:
      - dir: /vault/
        recursive: true
        action: Block
      - dir: /
        recursive: true
      - dir: /vault/
        recursive: true
        fromSource:
          - path: /bin/vault
  process:
    matchPaths:
      - path: /bin/busybox
      - path: /bin/vault
    action: Allow
```

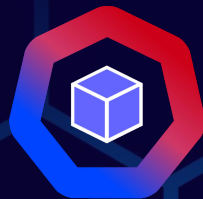
- Implement Least permissive access to sensitive assets



```
apiVersion: security.kubearmor.com/v1
kind: KubeArmorPolicy
metadata:
  name: conjur-oss
  namespace: conjur
spec:
  selector:
    matchLabels:
      container.name: '[conjur-oss]'
  action: Allow
  file:
    matchDirectories:
      - dir: /opt/conjur-server/
        recursive: true
        action: Block
      - dir: /opt/conjur-server/
        recursive: true
        fromSource:
          - path: /var/lib/ruby/bin/ruby
          - path: /usr/bin/bash
      - dir: /
        recursive: true
  process:
    matchDirectories:
      - dir: /var/lib/ruby/bin/
        recursive: true
  message: Conjur-oss policy
```

```
apiVersion: security.kubearmor.com/v1
kind: KubeArmorPolicy
metadata:
  name: conjur-postgres
  namespace: conjur
spec:
  selector:
    matchLabels:
      app: conjur-oss-postgres
  action: Allow
  file:
    matchDirectories:
      - dir: /etc/certs/
        recursive: true
        action: Block
      - dir: /etc/certs/
        recursive: true
        fromSource:
          - path: /usr/lib/postgresql/10/bin/postgres
      - dir: /var/lib/postgresql/data/
        recursive: true
        action: Block
      - dir: /var/lib/postgresql/data/
        recursive: true
        fromSource:
          - path: /usr/lib/postgresql/10/bin/postgres
      - dir: /
        recursive: true
  process:
    matchDirectories:
      - path: /usr/lib/postgresql/10/bin/postgres
  message: Conjur-Postgres-policy
```

- **AccuKnox tooling can help secure secrets**
 - At client locations where secrets are consumed
 - At secrets store where secrets are aggregated and kept
- **Least Permissive granular policy settings**
 - Provides Defense-In-Depth strategy {Process, File, Network}
- **Securing Secrets made easy by AccuKnox solution**



ACCUKNOX

SEE US IN ACTION

support@accuknox.com

Certified by

