KIALI-SECURITY-001 - Authentication bypass using forged credentials

Description

- **Disclosure date**: March 25, 2020
- **Affected Releases**: 0.4.0 to 1.15.0
- **Impact Score**: 9.4 - AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:H

A vulnerability was found in Kiali allowing an attacker to bypass the authentication mechanism. Currently, Kiali has four authentication mechanisms: *login, token, openshift* and *ldap*. All are vulnerable.

The vulnerability lets an attacker build forged credentials and use them to gain unauthorized access to Kiali.

Additionally, it was found that Kiali credentials were not being validated properly. Depending on the authentication mechanism configured in Kiali, this could facilitate unauthorized access into Kiali with forged and/or invalid credentials.

These vulnerabilities are filed as CVE-2020-1762 and CVE-2020-1764

Detection

Use the following bash script to check if you are vulnerable:

```
KIALI_VERSION=$(kubectl get pods -n istio-system -l app=kiali -o yaml | sed -n 's/^.*image: .*:v\(.*\)$/\1/p' | sort -u)
kubectl get deploy kiali -n istio-system -o yaml | grep -q LOGIN_TOKEN_SIGNING_KEY
TEST_KEY_ENV=$?
kubectl get cm kiali -n istio-system -o yaml | grep signing_key | grep -vq kiali
TEST_KEY_CFG=$?
VERSION_ENTRIES=(${KIALI_VERSION//./ })
echo "Your Kiali version found: ${KIALI_VERSION}"
[ ${VERSION_ENTRIES[0]} -lt "1" ] || ([ ${VERSION_ENTRIES[0]} -eq "1" ] && (\
  [ ${VERSION_ENTRIES[1]} -lt "15" ] || ([ ${VERSION_ENTRIES[1]} -eq "15" ] && ( \
  [ ${VERSION_ENTRIES[2]} -le "0" ])))) && echo "Your Kiali version is vulnerable"
[ $TEST_KEY_ENV -eq 1 ] && [ $TEST_KEY_CFG -eq 1 ] && echo "Your Kiali configuration looks vulnerable"
```

The script output will be similar to this:

```
Your Kiali version found: 1.14.0
Your Kiali version is vulnerable
Your Kiali configuration looks vulnerable
```

Mitigation

- Update to Kiali 1.15.1 or later.

Alternatively, if you cannot update to version 1.15.1, mitigation is possible by setting a secure signing key when deploying Kiali. If you installed via Kiali operator, you could use the following bash script:

```
SIGN_KEY=$(chars=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890; for i in {1..20}; do echo -n "${chars:RANDOM%${#chars}:1}"; done; echo)
kubectl get kiali -n $(kubectl get kiali --all-namespaces --no-headers -o custom-columns=NS:.metadata.namespace) -o yaml | sed "s/spec:/spec:\n    login_token:\n      signing_key: $SIGN_KEY/" |
```

If you installed via Istio helm charts or `istioctl` command, you could use the following bash script:

```
KIALI_INSTALL_NAMESPACE=istio-system
SIGN_KEY=$(chars=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890; for i in {1..20}; do echo -n "${chars:RANDOM%${#chars}:1}"; done; echo)
kubectl get cm kiali -n $KIALI_INSTALL_NAMESPACE -o yaml | sed "s/server:/login_token:\\n      signing_key: $SIGN_KEY\\n    server:/" | kubectl apply -f -
kubectl delete pod -l app=kiali -n $KIALI_INSTALL_NAMESPACE
```

Last modified October 6, 2021 : - restore News to top menu - update relnotes generator to generate markdown (b328cfa)